

VLSI Integrated Circuits and Systems: Principles and Design Methods

Olivier Sentieys

ENSSAT - Université de Rennes 1

IRISA/INRIA

sentieys@irisa.fr



Équipe-projet CAIRN
<http://www.irisa.fr/cairn>



ENSSAT
LANNION



Outline

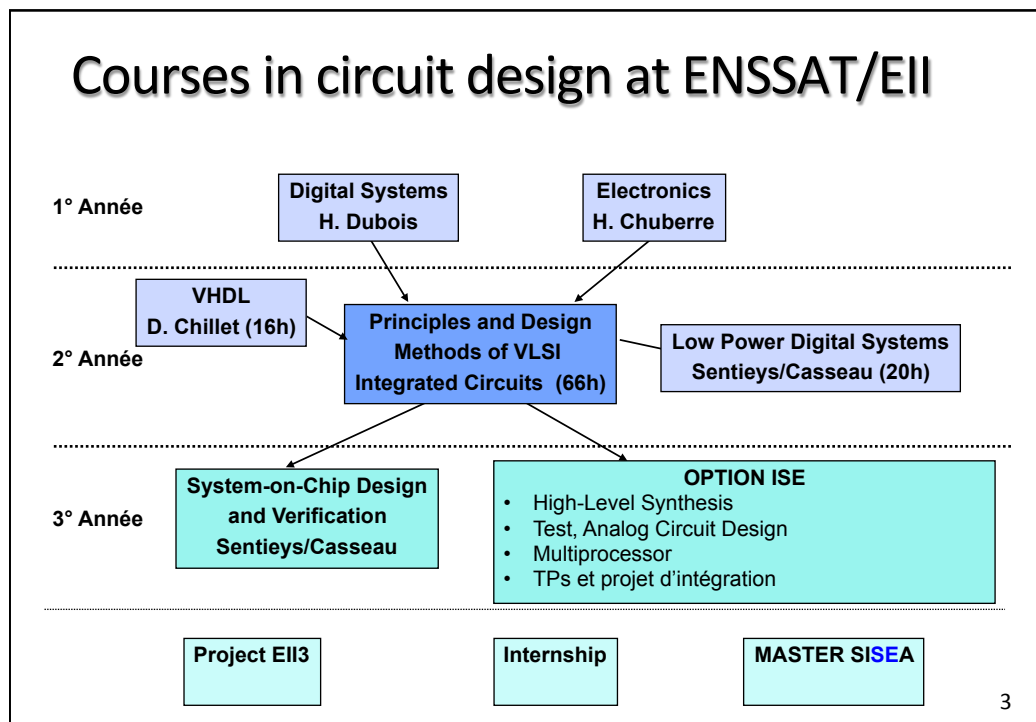
Principles and Design Methods of VLSI Integrated Circuits and Systems: from the idea to the chip

Introduction

- I. Integrated Circuit (IC) Technologies
- II. Design of CMOS Cells
- III. IC Design Methods
- IV. Synchronous Design of IC
- V. Logic Synthesis from VHDL
- VI. Power Estimation and Reduction
- VII. IC Design Project

http://people.rennes.inria.fr/Olivier.Sentieys/?page_id=95

2



Computer-Aided-Design (CAD) Tools for IC Design at ENSSAT

- Synopsys (Linux): logic synthesis (design compiler), power/timing analysis (prime time, power compiler)
- Cadence (Linux): IC place and route, analog and full-custom digital circuit design
- Altera Quartus (Linux/Windows): FPGA/CPLD design
- Xilinx ISE (Linux/Windows): FPGA/CPLD design
- MentorGraphics/ModelSim (Linux/Windows): HDL simulation and verification
- MentorGraphics/FPGA Advantage (Linux/Windows): FPGA design
- Cadence/Orcad/Pspice (Linux/Windows): Printed-Circuit-Board (PCB) design and verification
- MentorGraphics/Eldo (Linux): analog simulation

Detailed Outline (1/2)

I. Integrated Circuit (IC) Technologies

- MOS Technology
- (Bipolar Technology)
- IC Fabrication
- Silicon Technological Evolutions

II. Design of CMOS Cells

- Combinatorial Logic Cells
- Layout Design
- Sequential Logic Cells
- Delay and Power

III. IC Design Methods

- IC Classification
- Design Methods and CAD Tools
- IC Specification

5

Detailed Outline (2/2)

IV. Synchronous Design of IC

- Synchronous Design Rules and Principles
- Finite State Machine (FSM) plus Datapath Model
- Arithmetic Operators

V. Logic Synthesis from VHDL

- Methods and Tools
- Register-Transfer Level (RTL) Models using VHDL
- RT and Logic Synthesis CAD Algorithms

VI. Power Estimation and Reduction (now part of another course)

- Why Power is an Issue?
- Power Estimation
- Power Reduction

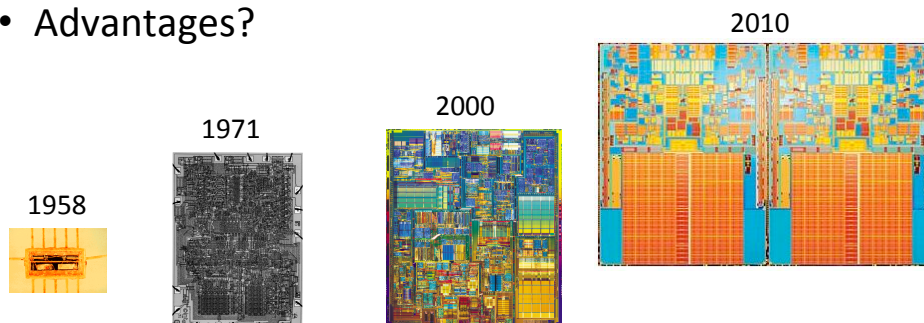
VII. IC Design Project

- It will be your turn to work!
- And to learn a lot about team-work!

6

Introduction

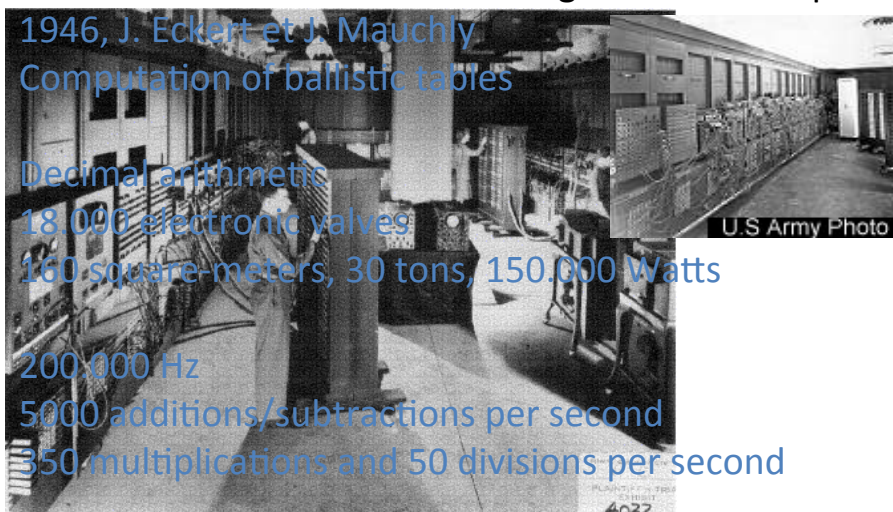
- A little bit of history and on the Moore law
- The Semi-Conductor Business
- ASIC, SOC et FPGA
- Advantages?



7

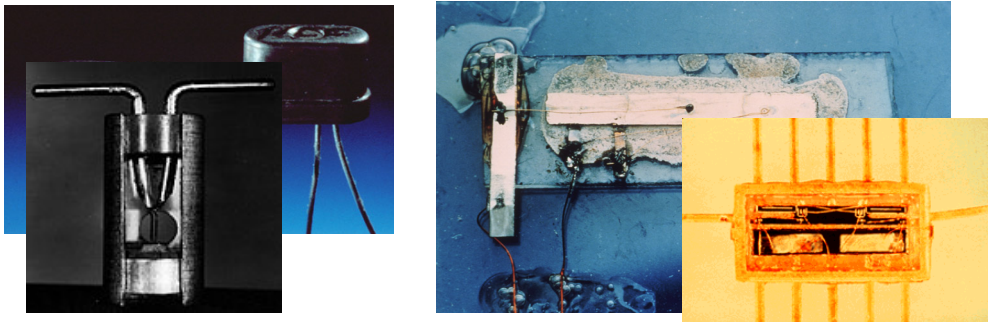
ENIAC – the first electronic computer

- ENIAC : Electronic Numerical Integrator and Computer
- 1946, J. Eckert et J. Mauchly
- Computation of ballistic tables
- Decimal arithmetic
- 18.000 electronic valves
- 160 square-meters, 30 tons, 150.000 Watts
- 200.000 Hz
- 5000 additions/subtractions per second
- 350 multiplications and 50 divisions per second



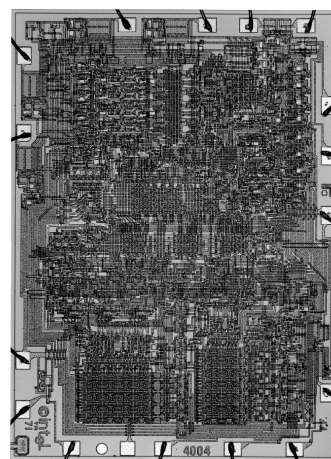
1958: The First Integrated Circuit

- 1947: W. Shockley (Bell Labs) invents the transistor (Nobel Price in 1956)
- 1958: J. Kilby (Texas Inst.) design the first IC (Nobel Price in 2000!)
 - Transistors, diodes, capacitors, wires assembled on a Silicon substrate
 - *"I perceived that a method for low-cost production of electronic circuits was in hand.... that instead of merely being able to build things smaller, we could fabricate entire networks in one sequence, and that we had extended the transistor's capability as a fundamental electronics tool."* Jack Kilby, 1958



The First Microprocessor

- Intel 4004
- 1971
- 400 kHz
- 4 bits
- 200 US\$ (1200FF)
- 0,06 MOPS
- 10 microns
- 2300 transistors
- 640 addressable bytes

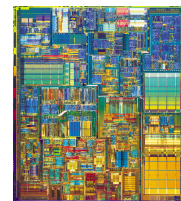


intel.

10

Then, everything will accelerate...

- 1970 4Kbits MOS Memory
- 1972 First Processor 4004 (Intel), NMOS technology
- 1977 16K DRAM et 4K SRAM produced
- 1979 64K DRAM produced
- 1980 Intel® first x86 Processor
- 1984 Intel® 80286 Processor (PC AT)
- 1986 1 megabit DRAM
- 1988 TI/Hitachi 16-megabit DRAM
- 1990 Intel® 80286 Processor with multimedia processing
- 1990 20 cm Wafer for production
- 1991 4 megabit DRAM produced
- 1993 Intel® Pentium® Processor
- 1997 Intel® Pentium® II Processor
- 1999 Intel® Pentium® III Processor
- 2000 Intel® Pentium® 4 Processor
- 2002 Intel® Itanium™ 2 Processor
- 2003 Intel® Pentium® M Processor, 1 gigabit DRAM
- ...



11

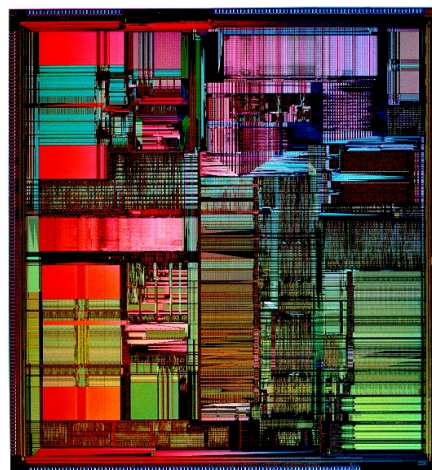
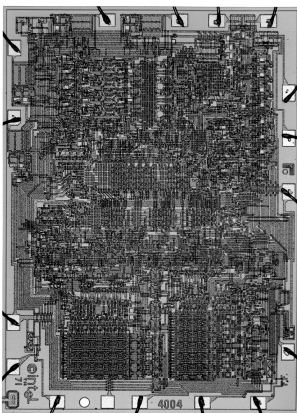
25 Years of Technological Advances

INTEL 4004 (1971)

4-bit data

2300 transistors, 10 microns

0,06 MOPS, 108 kHz



INTEL Pentium II (1996)

32-bit data

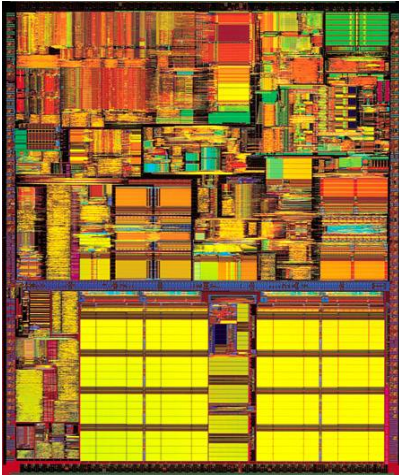
5.5M transistors, 0.35μ, 2 cm²

200 MHz, 200 MOPS, 3.3V, 35W

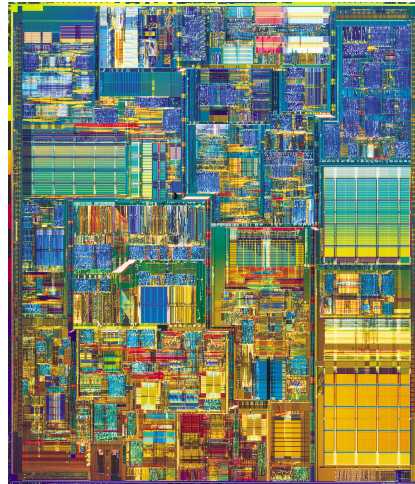
12

Intel' Microprocessor Gallery

1999 : Intel® Pentium® III Processor
9.5M Tr, 0.25um, 450MHz – 1GHz



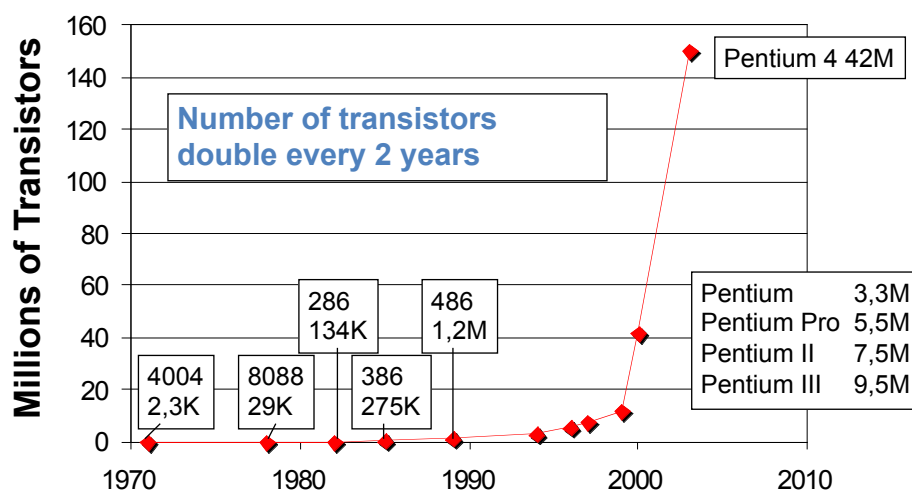
2000 : Intel® Pentium® 4 Processor
42M Tr, 0.18um, 1.5GHz – 3.6GHz



13

Number of Transistors

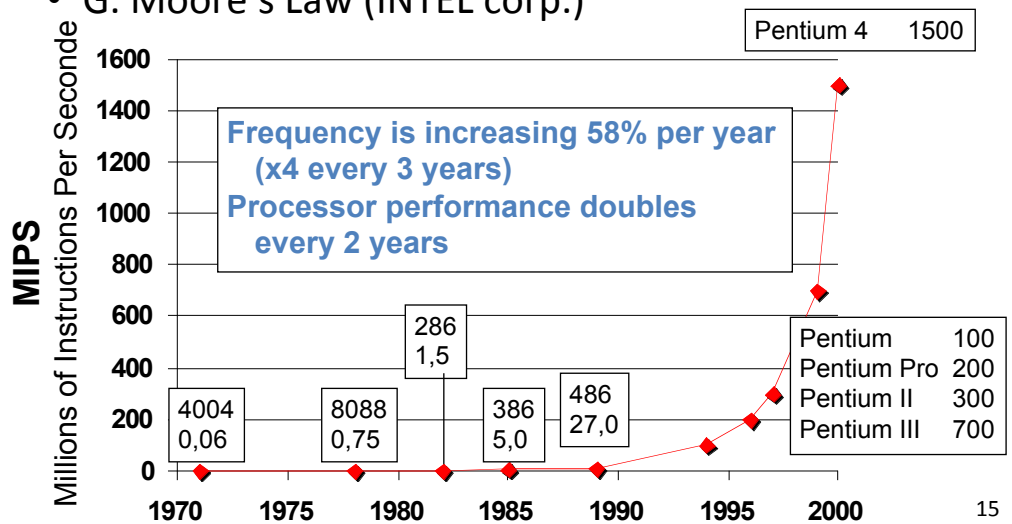
- G. Moore's Law (INTEL corp.)



14

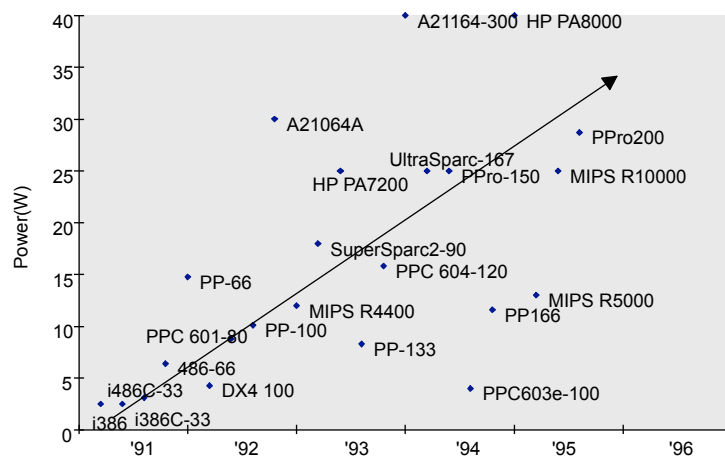
Clock Speed

- G. Moore's Law (INTEL corp.)



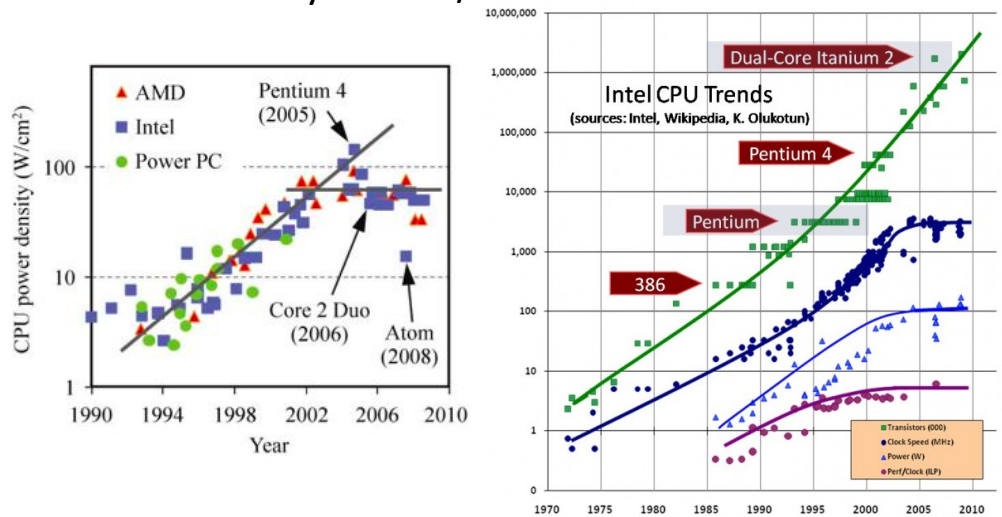
Why care about Power?

- Processor Power Evolution: x4 every 3 years



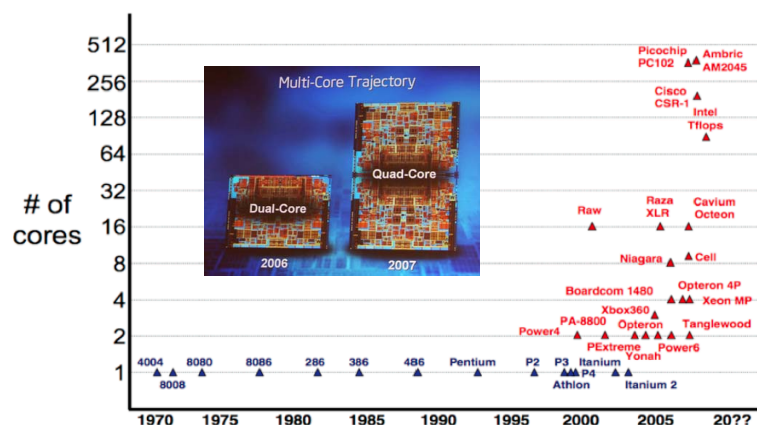
And then came the “Power Wall”

- Power Density: 100 W/cm² is a limit



and the “Multicore Era”

- Increasing performance by increasing # of cores



Semiconductor Market

- Semiconductor Industry Association (SIA) reported: *"The global semiconductor market hit a new record in 2006 with a sales volume of \$247.7 billion, up 8.9 percent from 2005"*
- Driven by consumer products such as cell-phones, MP3 players and HDTV receivers
- As the semiconductor industry completes one of its most successful years in 2010, worldwide semiconductor revenue is forecast to total \$314 billion in 2011, up 4.6% from 2010's (Gartner Inc.)

1H10 Top 20 Semiconductor Sales Leaders (\$M)

1H10 Rank	2009 Rank	Company	Headquarters	2009 Tot Semi	1Q10 Tot Semi	2Q10 Tot Semi	2Q/1Q % Change	1H10 Tot Semi
1	1	Intel	U.S.	32,325	9,485	9,910	4%	19,395
2	2	Samsung	South Korea	21,273	7,137	8,159	14%	15,296
3	3	TI	U.S.	9,697	2,990	3,262	9%	6,252
4	5	Toshiba	Japan	9,537	3,240	3,002	-7%	6,242
5	6	TSMC**	Taiwan	8,989	2,885	3,293	14%	6,178
6	4	Renesas Electronics*	Japan	9,649	2,787	2,821	1%	5,608
7	9	Hynix	South Korea	6,320	2,455	2,907	14%	5,262
8	7	ST	Europe	8,466	2,311	2,507	8%	4,818
9	10	Micron	U.S.	5,450	2,077	2,365	14%	4,442
10	15	Elpida	Japan	3,948	1,625	1,913	18%	3,538
11	8	Qualcomm***	U.S.	6,409	1,537	1,691	10%	3,228
12	11	AMD***	U.S.	5,403	1,574	1,653	5%	3,227
13	13	Infineon	Europe	4,617	1,438	1,540	7%	2,978
14	14	Broadcom***	U.S.	4,271	1,404	1,547	10%	2,951
15	12	Sony	Japan	5,245	1,300	1,248	-4%	2,548
16	16	NXP	Europe	3,547	1,079	1,105	2%	2,184
17	19	Freescale	U.S.	3,302	972	1,056	9%	2,028
18	24	UMC**	Japan	2,815	935	1,043	12%	1,978
19	17	MediaTek***	Taiwan	3,500	1,024	940	-8%	1,964
20	18	Fujitsu	Japan	3,377	955	933	-2%	1,888
—	—	Total	—	158,140	49,210	52,795	7%	102,005

*The merged entity of Renesas and NEC
Source: IC Insights, company reports

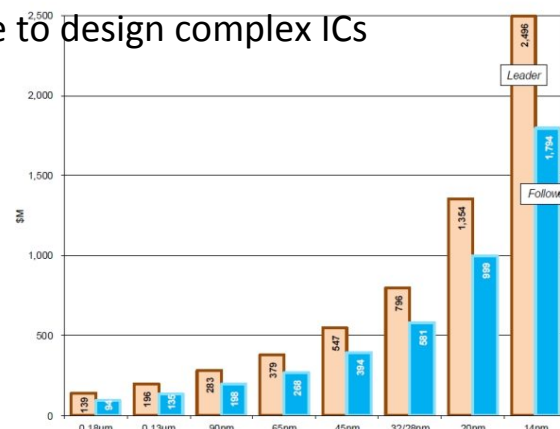
Foundry *Fabless

- 235 million units of PC were shipped in 2006,
- but more than 1 billion cellphones...
 - market is in DSP, MCU and memory

19

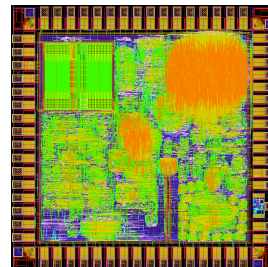
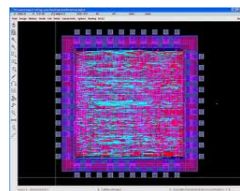
Semiconductor Business

- Foundry cost is high
 - lithography, test, packaging
 - investment increase of 19% every year
- More and more people to design complex ICs



What is an ASIC?

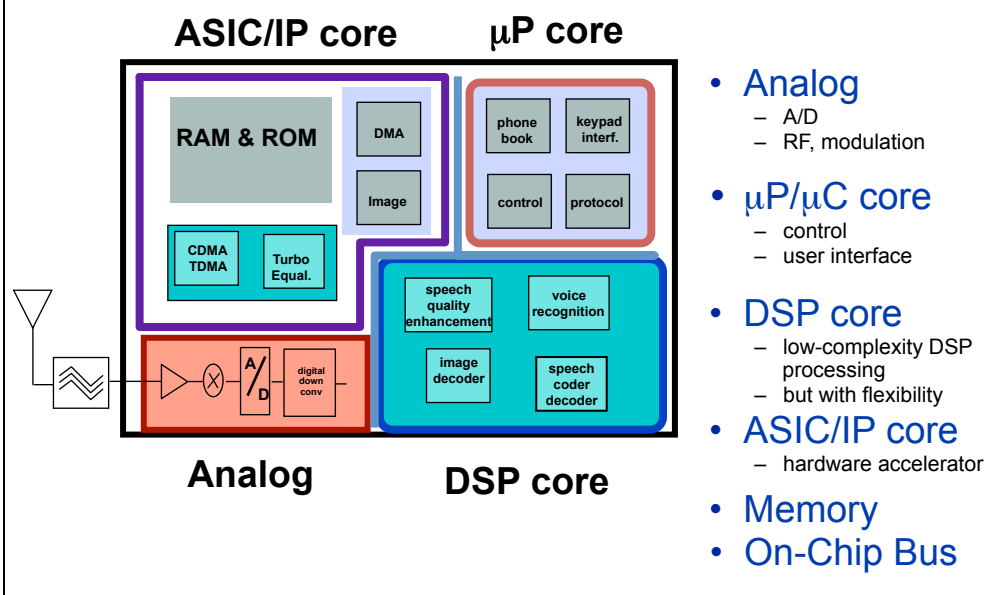
- ASIC: Application Specific Integrated Circuit
 - An ASIC is an integrated circuit (IC) customized for a particular use, rather than intended for general-purpose use
 - Examples: video recorder, camera, phone modem, speech processing ,etc.



OchreV2 IC designed at ENSSAT

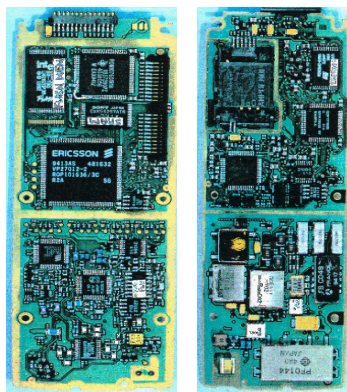
21

System-on-Chip (SoC)

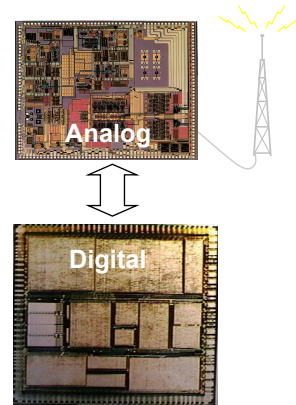


22

Ex. 1: 2G cellphone



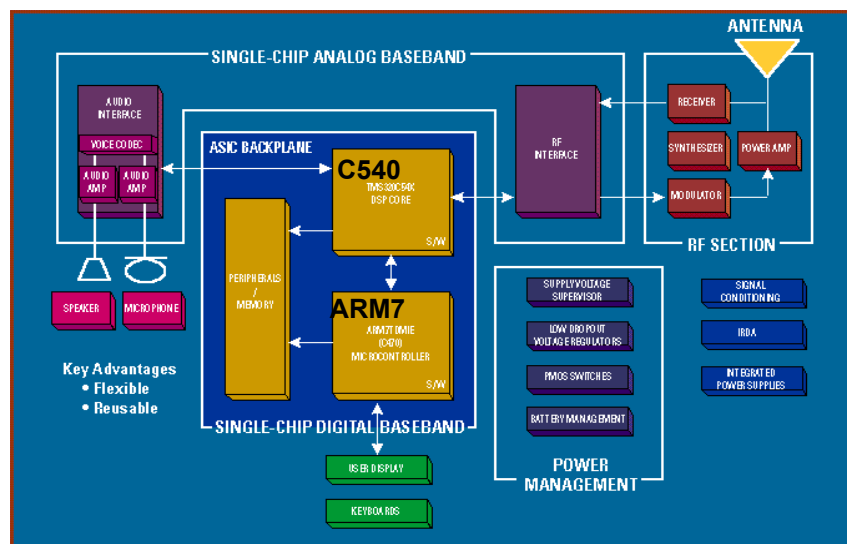
Your father's GSM



Now

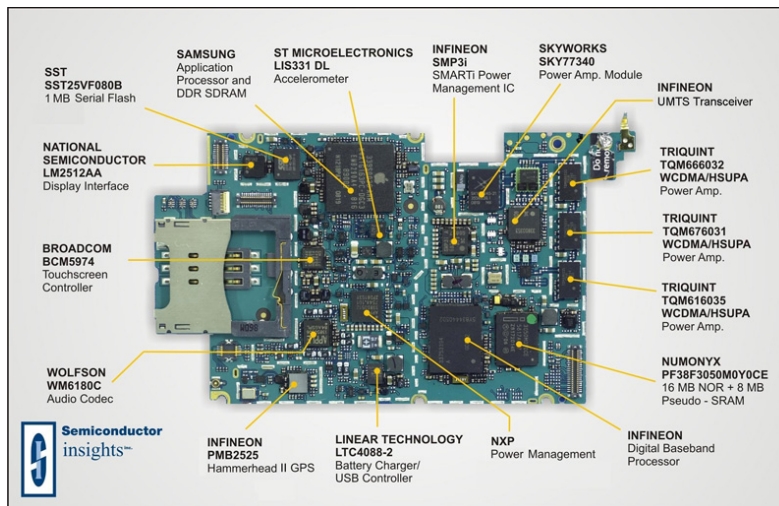
23

Ex. 1: 2G cellphone



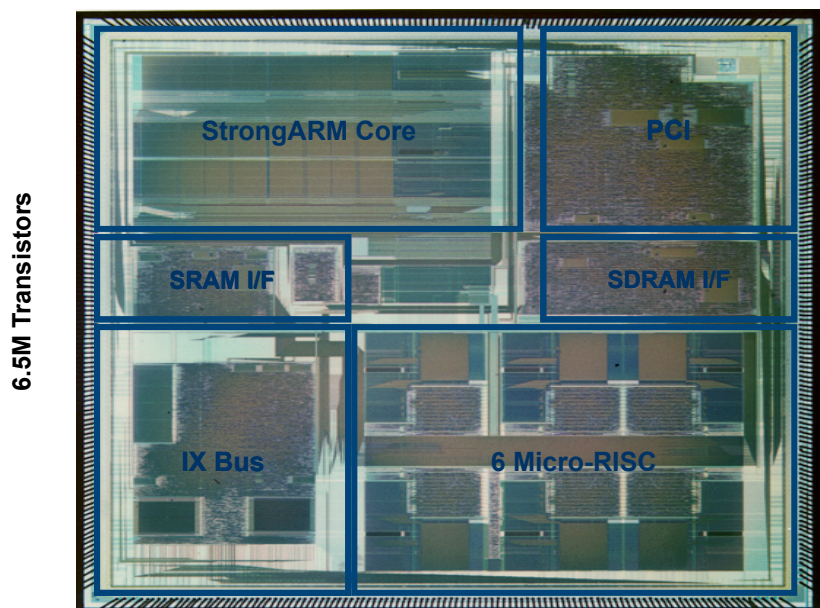
24

All is SoC Inside the iPhone!



25

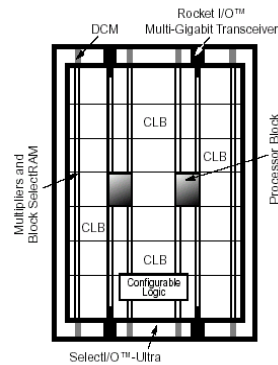
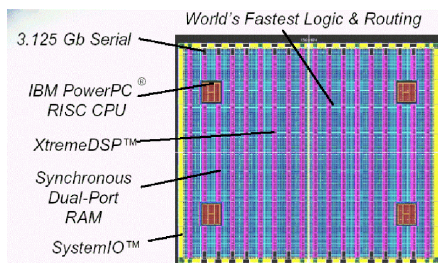
Ex. 2: IXP1200 Intel NPU



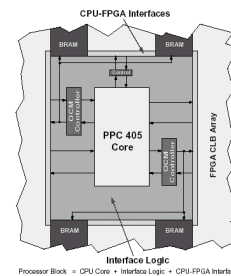
26

Ex. 3: FPGA Xilinx VIRTEX II Pro

- Up to 4 embedded processors IBM PowerPC 405, 300MHz
- Rocket I/O Multi-Gigaset Transceiver



Virtex II Pro Architecture



Embedded Processor



27

Ex. 3: FPGA Xilinx VIRTEX 7

Virtex-7 FPGAs
Optimized for Highest System Performance and Capacity
(1.0V, 0.9V)

	Part Number	XC7V285T	XC7V450T	XC7V585T	XC7V855T	XC7V1500T	XC7V2000T
Logic Resources	EasyPath™ Cost Reduction Solutions ⁽¹⁾	XCE7V285T	XCE7V450T	XCE7V585T	XCE7V855T	XCE7V1500T	XCE7V2000T
	Slices ⁽²⁾	44,700	70,450	91,050	133,350	229,050	305,400
	Logic Cells ⁽³⁾	286,080	450,880	582,720	853,440	1,465,920	1,954,560
	CLB Flip-Flops	357,600	563,600	728,400	1,066,800	1,832,400	2,443,200
Memory Resources	Maximum Distributed RAM (Kbits)	3,475	5,388	6,938	10,313	16,163	21,550
	Block RAM/FIFO w/ ECC (36Kbits each)	410	615	795	1,155	969	1,292
	Total Block RAM (Kbits)	14,760	22,140	28,620	41,580	34,884	46,512
Clock Resources	Mixed Mode Clock Managers (MMCM)	14	14	18	18	18	24
I/O Resources	Maximum Single-Ended I/O	700	700	850	850	850	1,200
	Maximum Differential I/O Pairs	336	336	408	408	408	576
	DSP48E1 Slices	700	980	1,260	1,800	1,620	2,160
Embedded Hard IP Resources	PCI Express Interface Blocks ⁽⁴⁾	2	3	3	3	3	4
	Analog Front End (XADC) / System Monitor Blocks	1	1	1	1	1	1
	Configuration AES / HMAC Blocks	1	1	1	1	1	1
	GTX 10.3125Gbps Transceivers	28	28	36	36	36	36
	GTH 13.1Gbps Transceivers	—	—	—	—	—	—
	GTZ 28.05Gbps Transceivers	—	—	—	—	—	—
Speed Grades	Commercial	-1L, -1, -2, -3	-1L, -1, -2, -3	-1L, -1, -2, -3	-1L, -1, -2, -3	-1L, -1, -2	-1L, -1, -2
	Industrial	-1L, -1, -2	-1L, -1, -2	-1L, -1, -2	-1L, -1, -2	-1L, -1	-1L, -1
Configuration	Configuration Memory (Mbits)	75.4	115.4	148.4	214.9	314.4	419.1
Package ⁽⁷⁾		Area		Available User I/O: 3.3V capable SelectIO™ Pins ⁽⁵⁾ , 1.8V SelectIO Pins ⁽⁶⁾ (GTX Transceivers)			

28

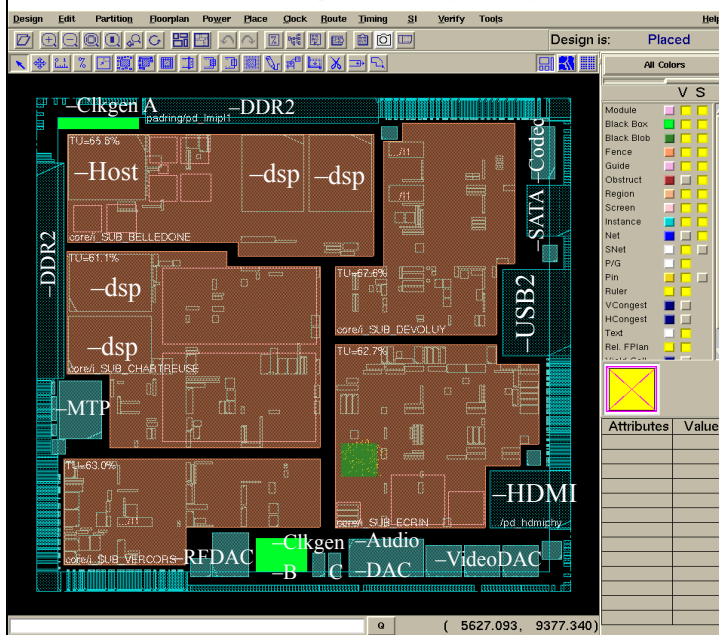
Ex. 4: Set Top Box (STb) STMicroelectronics

- STB Product is one chip solution for :
 - Dual H264-MPEG2-VC1 HD decoder, Triple TV display
 - MPEG2 [MP@HL](#)
 - ISO/IEC 14496-10/ITU Rec. H.264 Main profile level 4.1
 - VC1
 - Communications
 - 4 external transport streams (and three playbacks/timeshift from HDD or network)
 - 2 MII Ethernet, 3 USB2.0 and 2 SATA ports
 - Channel 3/4 mod
 - HD digital HDMI, 1 HD analog, 2 SD analog I/F
 - 1 Software modem including analog interface



29

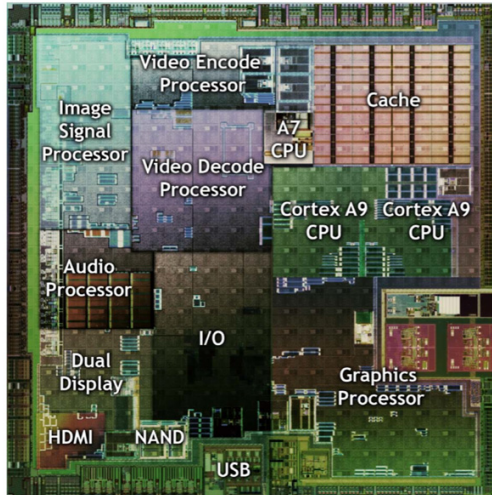
Ex. 4: Set Top Box (STb) STMicroelectronics



- 65nm LP 7ML
- 150Mtransistors
- 886 pads
- Top+5 BE partitions
- 18 FE subsystems
- 115 propagated clocks
- (19 for interconnect)
- 36 soft IPs
- 2 hard blocks
- 16 analog IPs
- 19 IOLIBs
- 29 internal blocks/glues
- 140 memory cuts

30

Ex. 5: NVIDIA Tegra 2 chip



- Smartphone market
- 40 nm, TSMC, 260M transistors
- Die size: 49mm²
- 2 ARM Cortex A9 1Ghz
- 1 ARM A7 (low-power)
- 300 Mhz (rest of the chip)
- Capable of processing images at a rate of 80 Mpixels/s
- Supports two cameras (12+5 Mpixels)
- Real time H.264 video encoding
- Decode 1080p H.264 videos at up to 20Mbps

31

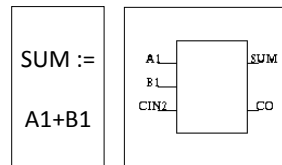
Advantages and Drawbacks

- Advantages of ASIC/FPGA solutions
 - Size and Power are reduced while Speed is increased: **Energy Efficiency**
 - Cost may be reduced: depends on volume and on ASIC/FPGA choice
 - Confidentiality, Technological Advantage
- Drawbacks of ASIC/FPGA
 - Cost of the first produced chip is more than 1M\$
 - Not true for FPGA
 - Requires highly-qualified people (but you will be there for that...)
 - Design Complexity: an SoC is HW + SW
 - Verification is a nightmare!
 - Partial loss of control in delays and costs; Higher risk
 - Partial unpredictability
 - Risk and cost overheads in case of design errors in the first prototype

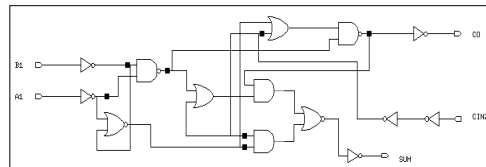
32

Design Flow: from algorithm to circuit

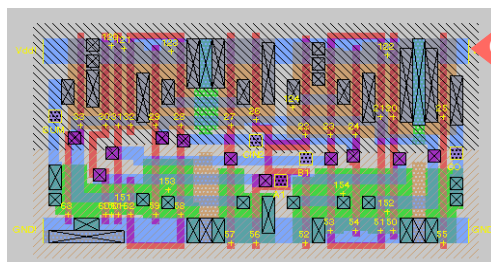
Register Transfer Level (RTL)



Gate Level

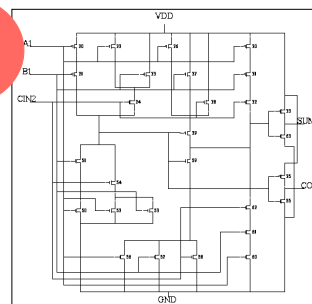


Algorithm



Layout Level

Circuit



Transistor or Circuit Level

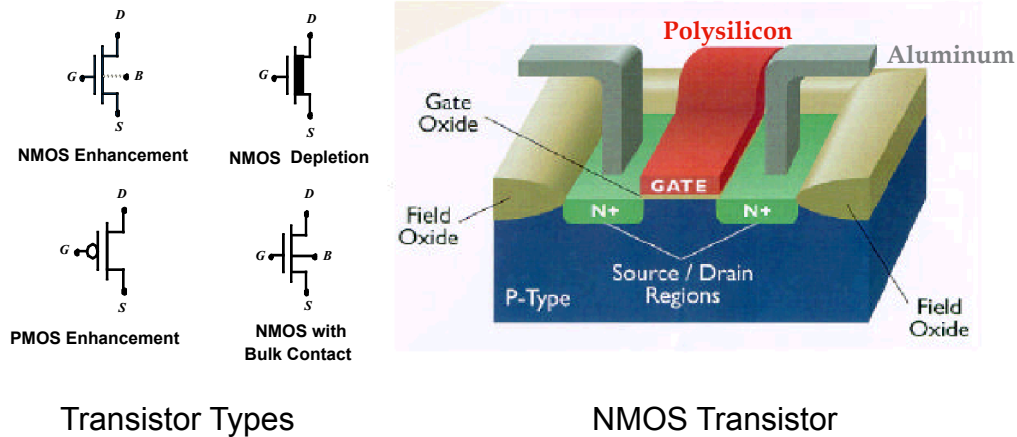
33

I. Integrated Circuit (IC) Technologies

1. MOS Technology
 1. The MOSFET Transistor
 2. Transistor Performance Models
 3. MOS Technologies (nMOS, pMOS, CMOS)
2. (Bipolar Technology)
3. IC Fabrication
 1. Fabrication Process
 2. Example of a Diode and a MOSFET
4. Silicon Technological Evolutions
 1. Processor and Technology Roadmaps
 2. Technology Scaling
5. Interconnects

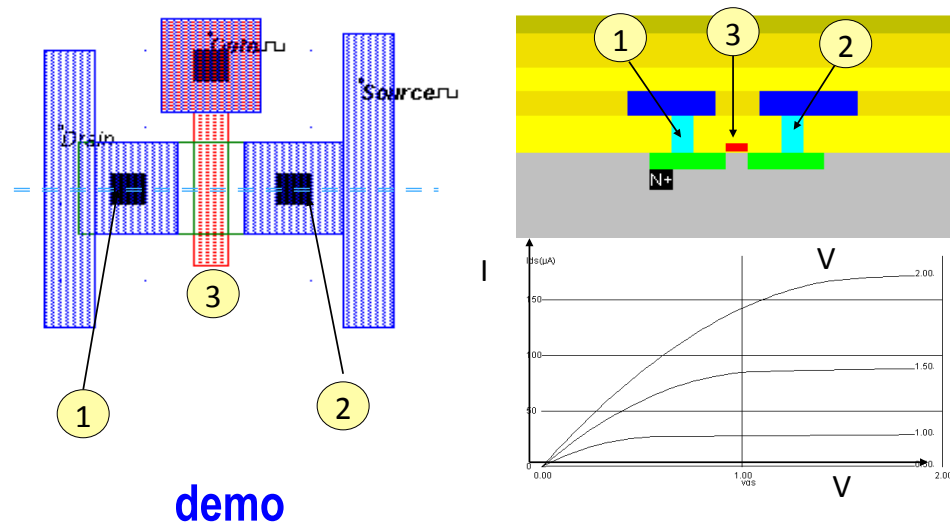
34

1.1 MOSFET: Metal–Oxide–Semiconductor Field-Effect Transistor



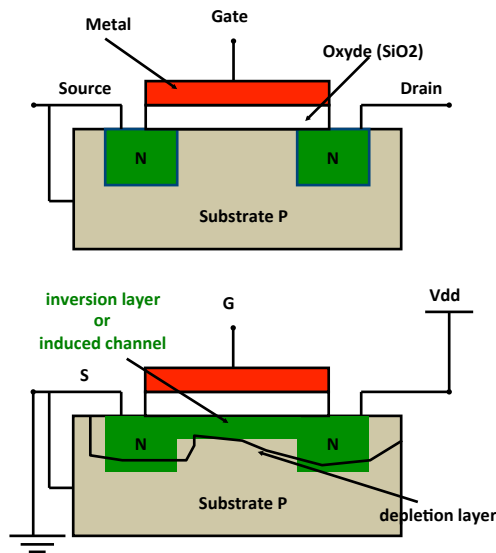
35

1.1 MOS Transistor



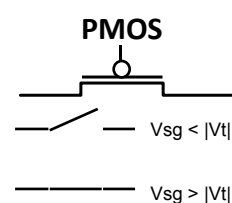
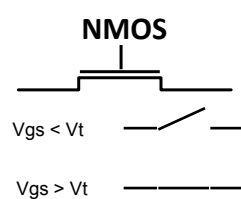
36

1.1 NMOS Transistor



- MOSFET: Metal (Polysilicium) Oxide Silicon Field Effect Transistor
- Cutoff or sub-threshold mode:
 $V_{GS} < V_t$, $R_{SD} \approx +\infty$
- Linear mode:
 $V_{GS} > V_t$ and $V_{DS} < V_{GS} - V_t$
 – a channel is created which allows current to flow between the drain and the source
- Saturation mode:
 $V_{GS} > V_t$ and $V_{DS} > V_{GS} - V_t$ 37

NMOS/PMOS Transistors



Voltage Level Transfer (Source > Drain)

- A '0' is well transmitted
- A degraded '1' is transmitted ($V_{dd} - V_t$)

- A '1' is well transmitted
- A degraded '0' is transmitted ($V_{ss} + |V_t|$)

Charge Carriers

Electrons

Holes

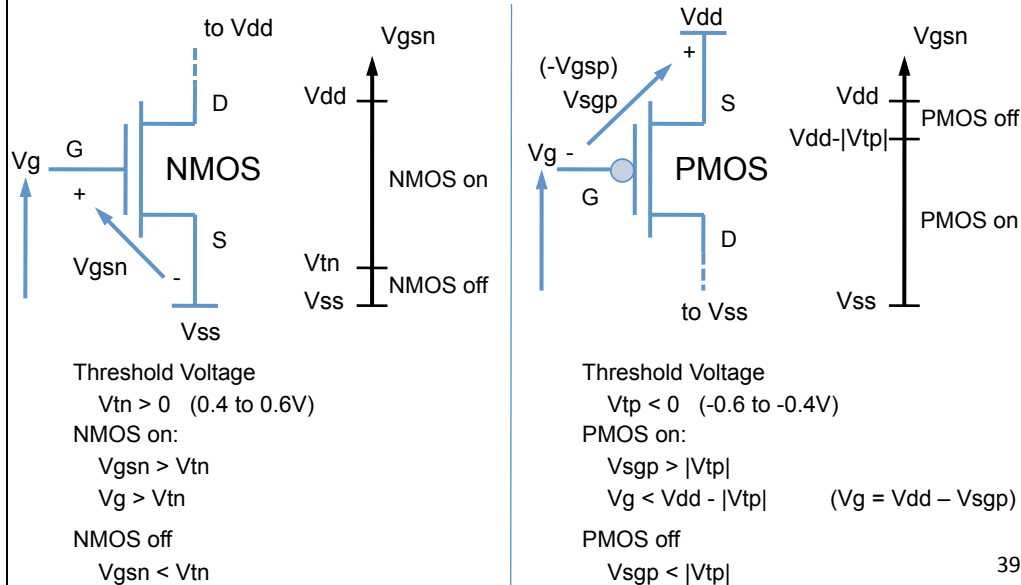
Substrate/Bulk/Well Voltage

Vss

Vdd

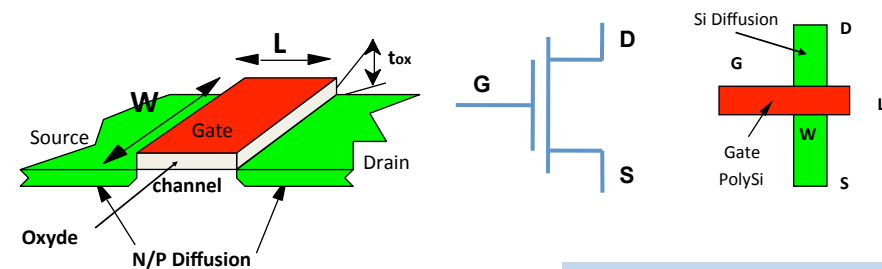
38

NMOS/PMOS Transistors



39

1.2 MOS Transistor Models



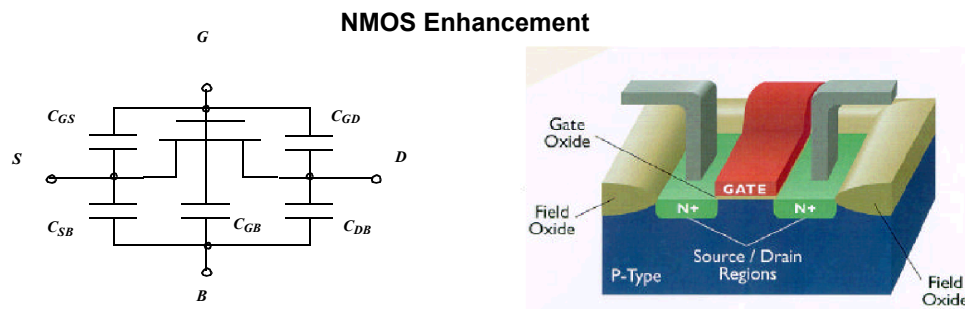
$$I_{ds} = \begin{cases} 0 & \text{cut-off} & V_{GS} < V_t \\ K \left((V_{GS} - V_t) V_{DS} - \frac{V_{DS}^2}{2} \right) & \text{linear} & 0 < V_{DS} < V_{GS} - V_t \\ \frac{K}{2} (V_{GS} - V_t)^2 & \text{saturated} & 0 < V_{GS} - V_t < V_{DS} \end{cases}$$

- K is a function of W/L
- $I_{ds \max}$ depends on W
- Temperature $\uparrow$: influence on μ and $I_{ds \max}$?

- W : gate width
- L : gate length
- tox : oxide width ($\#L/10$)
- Cox : gate oxide capacitance per unit area
- $K = \mu \cdot \epsilon \cdot W / (tox \cdot L) = \mu \cdot Cox \cdot W/L$
- μ : charge-carrier effective mobility
 - NMOS (electrons) $\mu_N = 500 \text{ cm}^2/\text{V-sec}$ # 2 μP
 - PMOS (holes) $\mu_p = 270 \text{ cm}^2/\text{V-sec}$
- ϵ : oxide permittivity # 4 $\epsilon_0 = 3.5 \cdot 10^{-13} \text{ F/cm}$

40

NMOS Parasistic Elements



$$\text{Drain-Source Resistance: } R_{DS} = \frac{1}{K(V_{dd} - V_t)} = \frac{1}{k(V_{dd} - V_t)} \frac{L}{W}$$

$$\text{Gate Capacitor: } C_g = \frac{\epsilon W L}{t_{ox}} = W L C_{ox}$$

$$\text{Drain/Source-Bulk Capacitor: } C_{sb} = C_{db} \approx W L C_j$$

$$\text{Delay: } \tau = R_{DS} C_g = \frac{L^2}{\mu(V_{dd} - V_t)}$$

41

MOS SPICE model 1

- Expression of I_{ds}

Mode	Condition	Expression
Cut - off	$V_{gs} < 0$	$I_{ds} = 0$
Linear	$V_{ds} < V_{gs} - V_t$	$I_{ds} = KP \frac{W}{L} \left((V_{gs} - V_t) V_{ds} - \frac{V_{ds}^2}{2} \right)$
Saturated	$V_{ds} > V_{gs} - V_t$	$\frac{KP W}{2 L} (V_{gs} - V_t)^2$

- Threshold Voltage

$$V_t = VT0 + GAMMA + \sqrt{PHI - Vb} - \sqrt{PHI}$$

Parameter	Definition	NMOS 0.25u	PMOS 0.25u
VT0	Threshold Voltage	0.4V	-0.4V
KP	Transconductance Coefficient	300 μ A/V ²	120 μ A/V ²
PHI	Surface Inversion Potential	0.3V	0.3V
GAMMA	Bulk Threshold Parameter	0.4V ^{0.5}	0.4V ^{0.5}
W	Channel Width	0.5-20 μ m	0.5-40 μ m
L	Channel Length	0.25 μ m	0.25 μ m

42

MOS SPICE model 3

Cut - Off $V_{gs} < 0$ $I_{ds} = 0$

Normal $V_{gs} > V_{on}$

$$I_{ds} = K_{eff} \frac{W}{L_{eff}} (1 + KAPPA \cdot V_{ds}) V_{de} \left((V_{gs} - V_t) - \frac{V_{de}}{2} \right)$$

$$V_{on} = 1.2V_t, \quad V_t = V_{TO} + GAMMA \left(\sqrt{PHI - V_b} - \sqrt{PHI} \right)$$

$$V_{de} = MIN(V_{ds}, V_{dsat}), \quad V_{dsat} = V_c + V_{sat} - \sqrt{V_c^2 + V_{sat}^2}, \quad V_{sat} = V_{gs} - V_t$$

$$V_c = V_{MAX} \frac{L_{eff}}{0.06}, \quad L_{eff} = L - 2 \cdot LD, \quad K_{eff} = \frac{KP}{1 + THETA(V_{gs} - V_t)}$$

Sub - Threshold $V_{gs} < V_{on}$

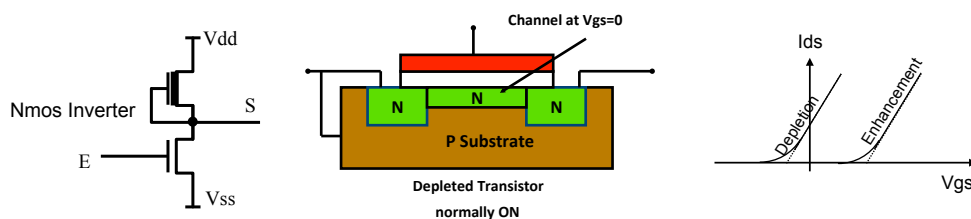
$$I_{ds} = K_{eff} \frac{W}{L_{eff}} (1 + KAPPA \cdot V_{on}) V_{de} \left((V_{gs} - V_t) - \frac{V_{de}}{2} \right) e^{\frac{q(V_{gs} - V_{on})}{nkT}}$$

$$V_{de} = MIN(V_{on}, V_{dsat})$$

Parameter	Definition	NMOS 0.25u	PMOS 0.25u
LD	Lateral diffusion into channel	0.01μm	0.01μm
KAPPA	Saturation field vector	0.01V ⁻¹	0.01V ⁻¹
VMAX	Maximum drift velocity	150km/s	150km/s
THETA	Mobility degradation factor	0.3V ⁻¹	0.3V ⁻¹
NSS	Subthreshold factor	0.07V ⁻¹	0.07V ⁻¹ ⁴³

1.3 MOS Technologies

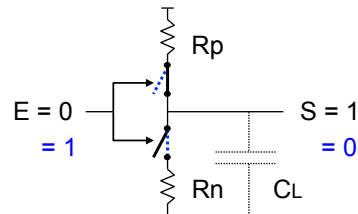
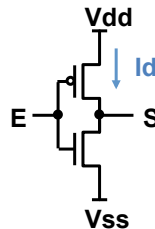
- NMOS (or PMOS) Technologies (1970-1980)
 - Only one type of transistor NMOS or PMOS
 - Resistances made with depleted NMOS transistors



- Good integration, but...
- Rising/Falling delays unbalanced
- '1' level is degraded
- Static power consumption when S='0'

1.3 MOS Technologies

• CMOS Technology



– Perfect Voltage Level Transmission

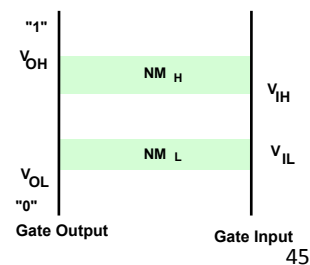
- PMOS transmits Vdd ('1') when E=0
- NMOS transmits Vss ('0') when E=1

– Noise Margin:

- noise level at input without modifying logic level

– Excellent Noise Margins

- $V_{OH}=V_{dd}$; $V_{OL}=V_{ss}$



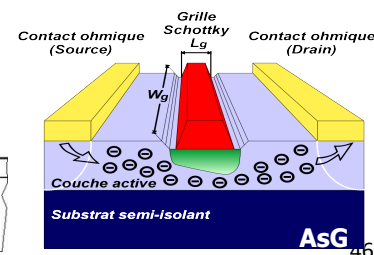
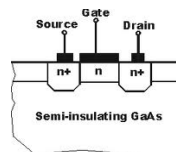
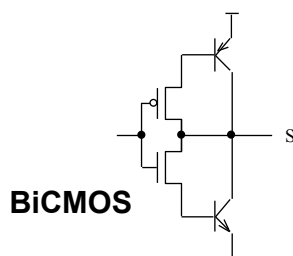
1.3 MOS Technologies

• BiCMOS

- Combine advantages of bipolar (speed) and MOS (density, power)
- Push-Pull bipolar amplifier at the output of the cell

• Gallium Arsenide (GaAs) Technology

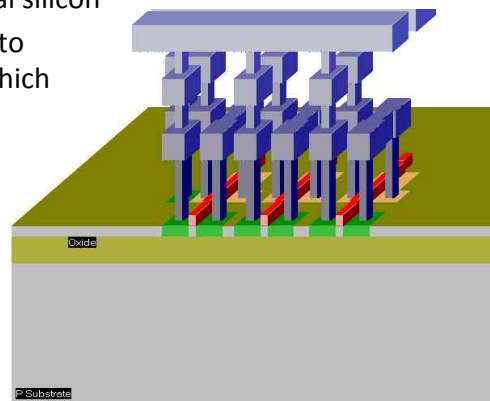
- Compound of the elements gallium and arsenic, a III/V semiconductor
- Higher electron mobility than Si: high frequency
- No P-channel MESFET
- Used in RF, LEDs or lasers



1.3 MOS Technologies

- SOI (Silicon On Insulator) Technology

- In Bulk Silicon CMOS substrate and well imply parasitic capacitors and leakage currents
- Use of a layered silicon-insulator-silicon substrate (silicon dioxide or sapphire) in place of conventional silicon
- Lower parasitic capacitance due to isolation from the bulk silicon, which improves power consumption
- Resistance to latchup due to complete isolation of the n- and p-well structures
- Compatible with CMOS
- Integration density is higher



I. Integrated Circuit (IC) Technologies

1. MOS Technology

1. The MOSFET Transistor
2. Transistor Performance Models
3. MOS Technologies (nMOS, pMOS, CMOS)

2. (Bipolar Technology)

3. IC Fabrication

1. Fabrication Process
2. Example of a Diode and a MOSFET

4. Silicon Technological Evolutions

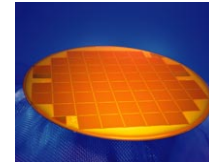
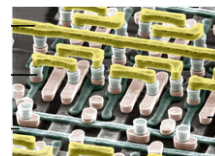
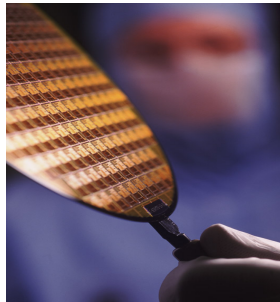
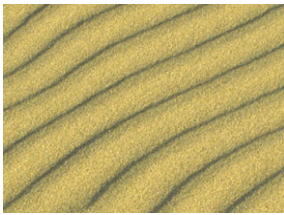
1. Processor and Technology Roadmaps
2. Technology Scaling

5. Interconnects

48

I.2 How ICs are made?

From sand to silicon
From silicon to integrated circuit



<http://www.intel.com/education/makingchips>

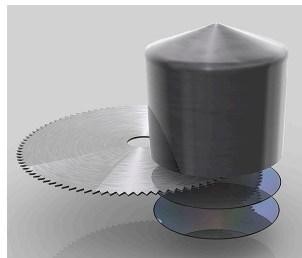
49

Integrated Circuit Fabrication

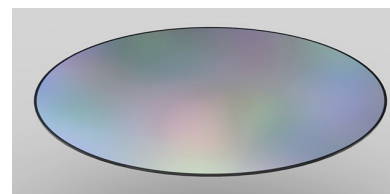
- Silicon ingot (#100kg) pure at 99,9999999%



**Mono-crystal
Silicon Ingot**



Ingots slicing

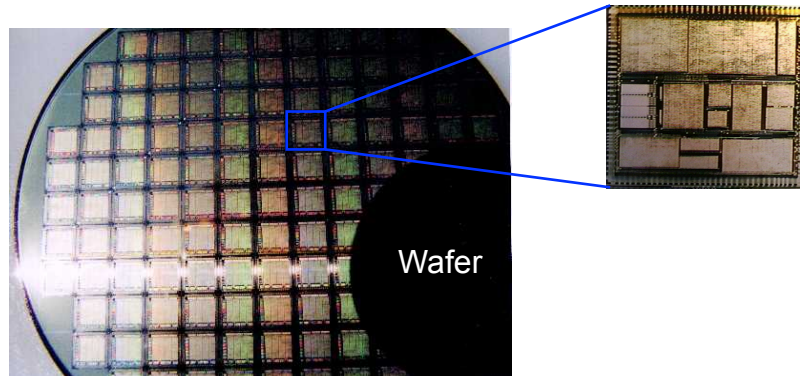


Silicon Wafer

50

Integrated Circuit Fabrication

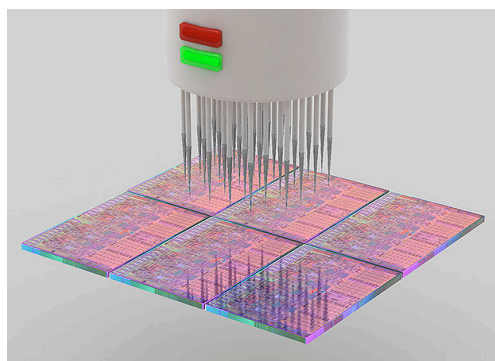
- Wafer: is a thin slice of Si material (substrate)
 - 450 mm (17.7 inch), thickness 925 μm
- Set of identical chips (die) printed on a Wafer



51

Integrated Circuit Fabrication

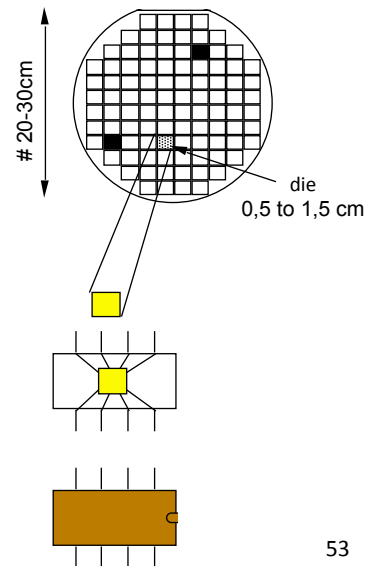
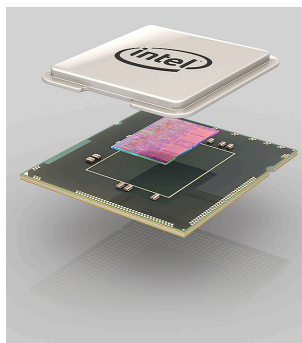
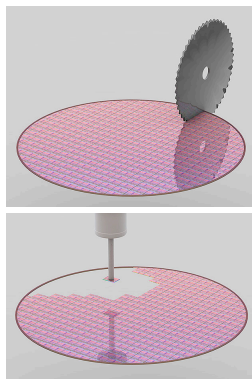
- Wafer testing of chips using a wafer prober



52

Integrated Circuit Fabrication

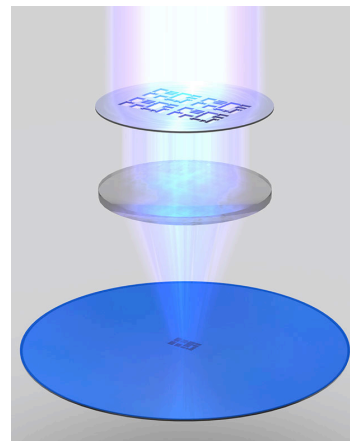
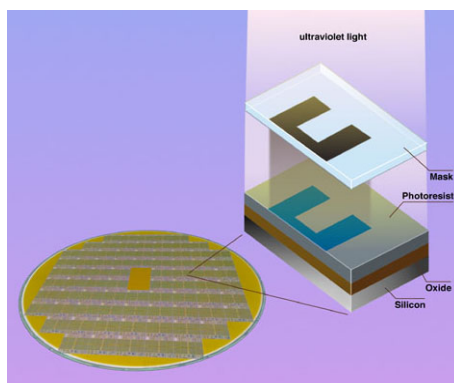
- Wafer is then cut into *dies*
 - Dicing
- Packaging
- Final testing with package



53

Integrated Circuit Fabrication

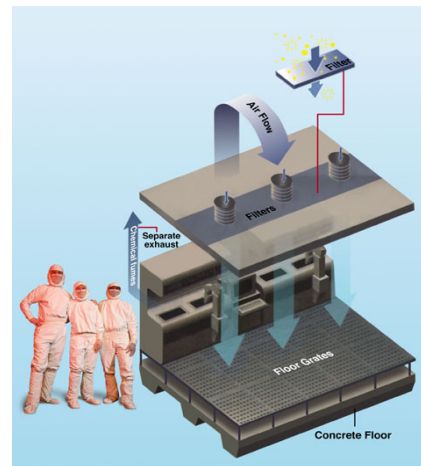
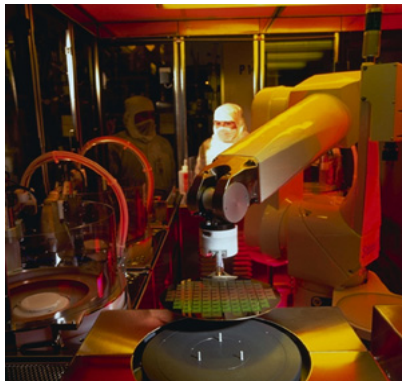
- Photolithography is like printing a tiny 3D book



54

Integrated Circuit Fabrication

- Extremely clean operating conditions

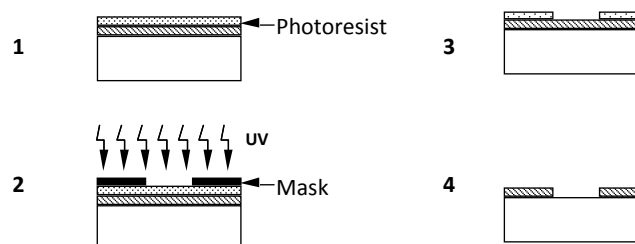


55

Lithography

Start with a wafer with SiO_2 (surface oxidation)

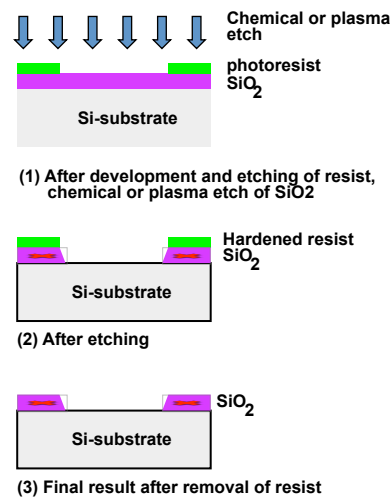
1. Spin on a photoresist
2. Pattern photoresist with a mask
3. Etching of photoresist
4. Wash off photoresist



56

Etching

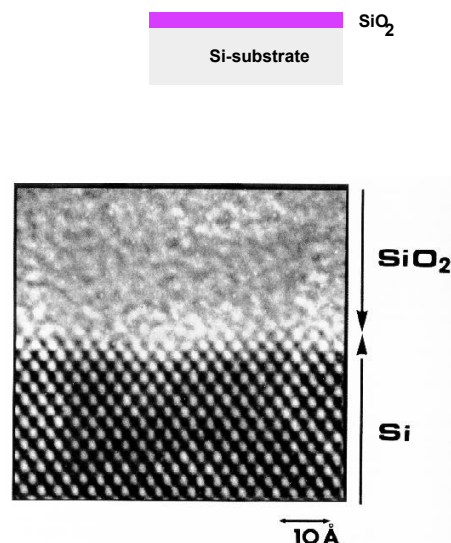
- Pattern material on the surface
- Desired shape is patterned with photoresist through a mask
- Chemical (wet) or plasma (dry) etching of the material



57

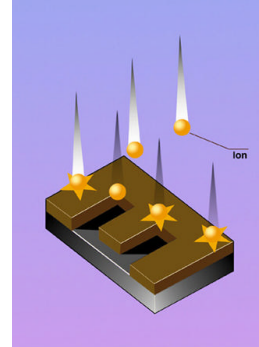
Oxide Deposit/Growth

- Oxidation of the silicon surface creates a SiO₂ layer that acts as an insulator
- Oxide layers are also used to isolate metal interconnections
- Wafer is placed in a high-temperature furnace to make the silicon react with oxygen or water vapor
- $\text{Si} + 2\text{H}_2\text{O} \rightarrow \text{SiO}_2 + 2\text{H}_2$
- PVD: Physical Vapor Deposition
- CVD: Chemical Vapor Deposition



Ion Implantation

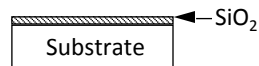
- Ion implantation is used to add doping materials to change the electrical characteristics of silicon locally
- Dopant ions penetrate the surface, with a penetration depth that is proportional to their kinetic energy
- P+ implant: boron
- N+ implant: arsenic or phosphorous



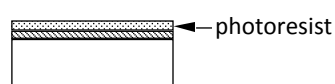
59

Example: Diode Fabrication

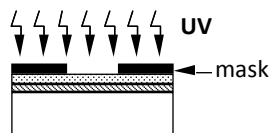
a) Oxide deposit



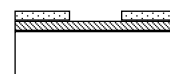
b) Spin on photoresist



c) UV light exposure through mask



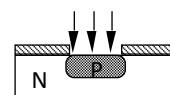
d) Pattern photoresist



e) Pattern SiO2



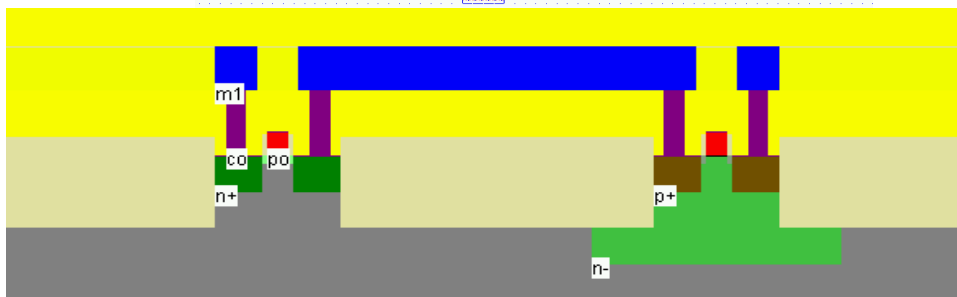
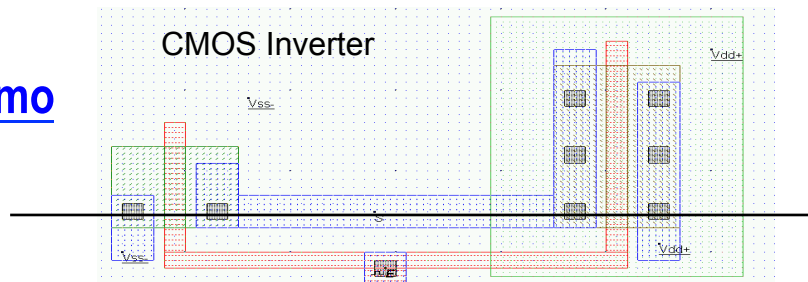
f) Ion implant (P-type)



60

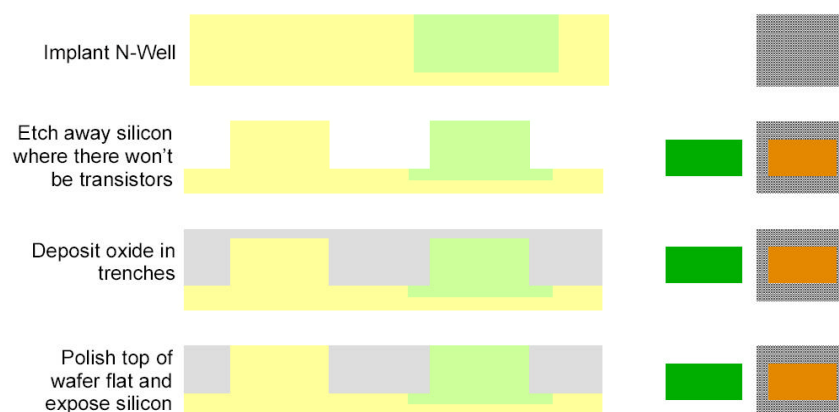
CMOS Inverter

demo



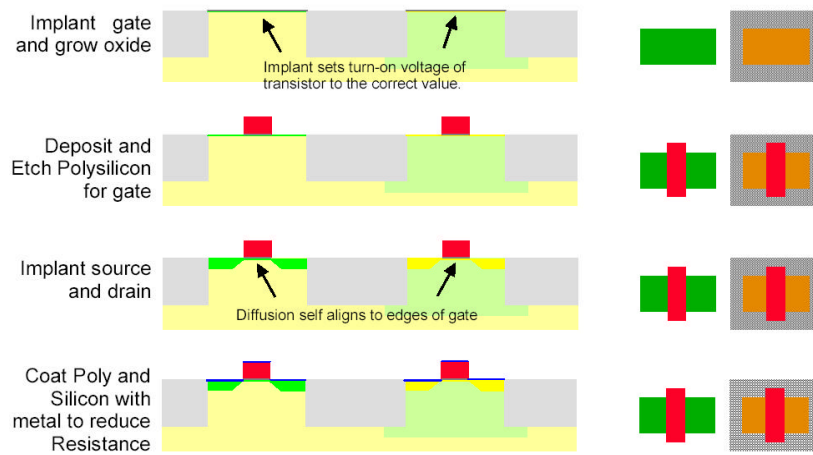
61

CMOS Gate Fabrication



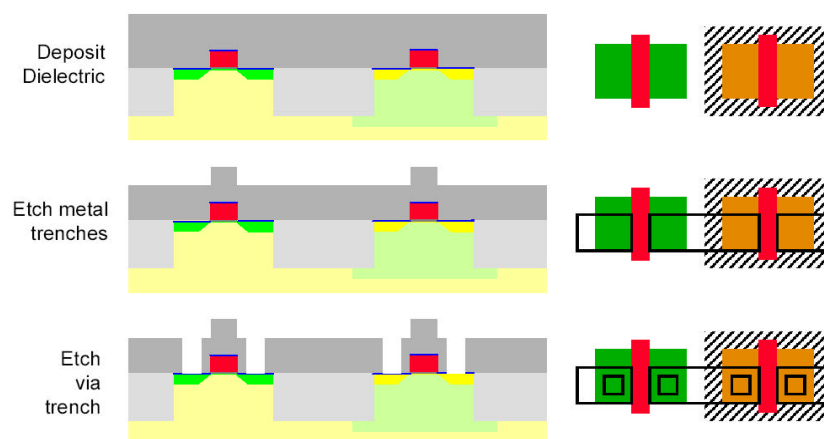
62

CMOS Gate Fabrication



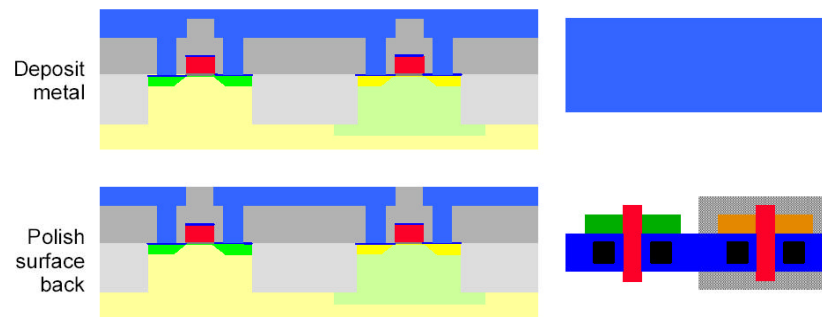
63

CMOS Gate Fabrication



64

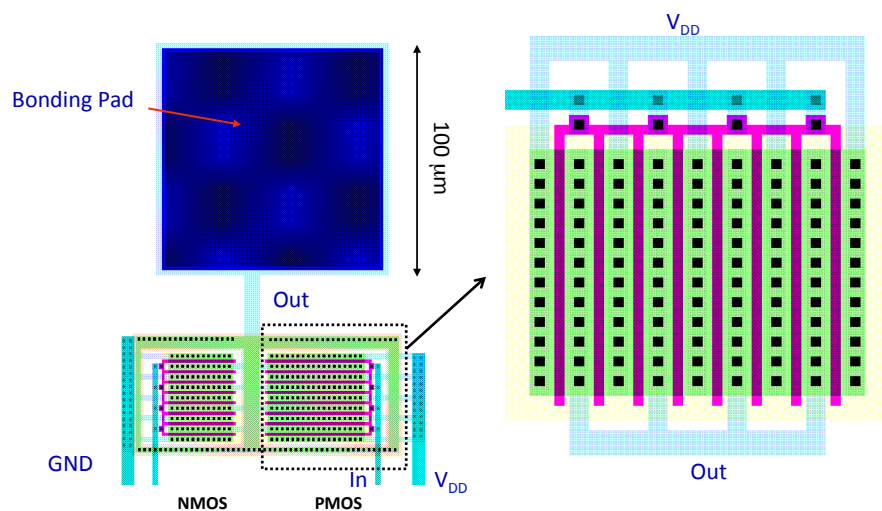
CMOS Gate Fabrication



65

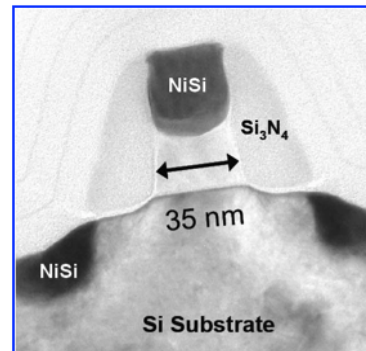
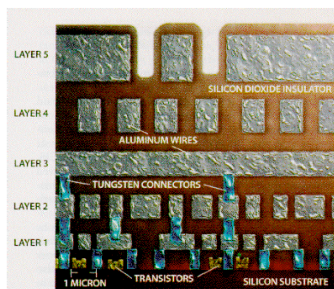
I/O Pads

- I/O pads use large transistors



66

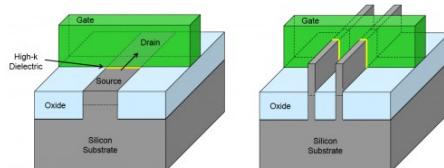
Real Transistors and Metal Layers



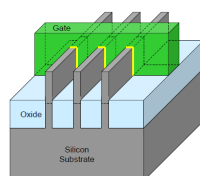
67

3D FinFET Transistors (Intel) ...

- At 22nm transistors go 3D



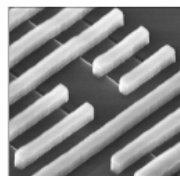
22 nm Tri-Gate Transistor



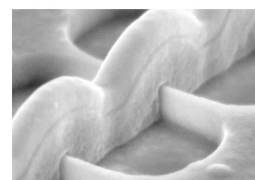
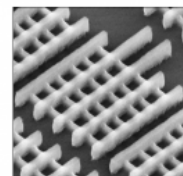
Tri-Gate transistors can have multiple fins connected together to increase total drive strength for higher performance



32 nm Planar Transistors



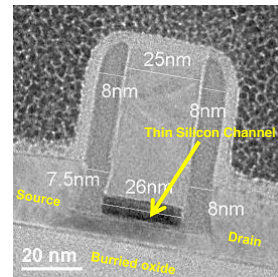
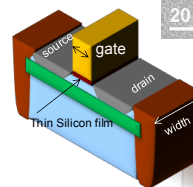
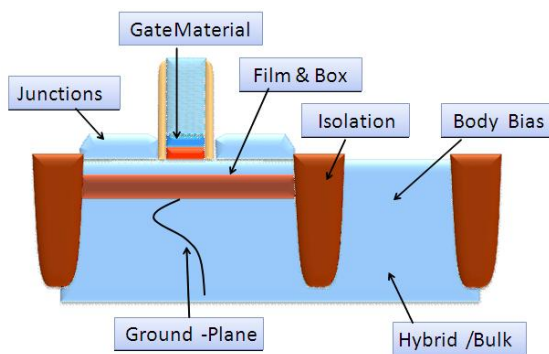
22 nm Tri-Gate Transistors



68

fight against FDSOI Transistors (ST)

- Ultra Thin Body (FD) – SOI
 - Total dielectric isolation
 - Lower S/D capacitances & leakages
 - Latch-up immunity
 - Improved VT variation

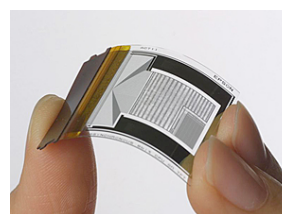
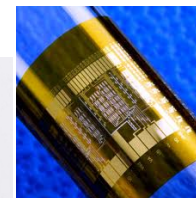
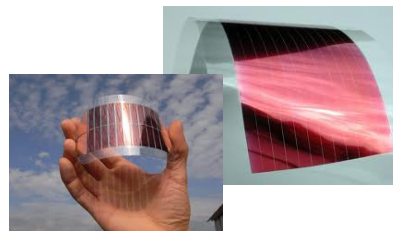


STMicro's roadmap

69

Flexible Electronics

- Plastic substrate
 - Organic PV cells
 - MOS Transistors
 - polysilicon thin-film transistors (LTPS-TFTs) on a plastic substrate



World's First Flexible 8-Bit Asynchronous Microprocessor

Table 1. Comparison of CMOS and thin-film transistor (TFT) device technologies.

Technology parameter	Device technology			
	CMOS	Amorphous Si TFT	Metal-oxide TFT	SAM organic TFT
Process temperature	1,000°C	250°C	<100°C	<100°C
Process technology	Lithography	Lithography	Roll-to-roll lithography	Shadow mask
Feature size	32 nm	8 µm	5 µm	50 µm
Substrate	Water	Glass or plastics	Glass or plastics	Water or plastics
Device type	Complementary	N-type only	N-type only	Complementary
Supply voltage	1 V	20 V	20 V	1 V
Mobility (cm ² /Vs)	1,500	1	10	0.01 or 0.5 (n- or p-type)
Cost/Area	High	Medium	Low	Low
Lifetime	Very good	Good	Good	Medium

Table from: Huang et al., Robust Circuit Design for Flexible Electronics, IEEE Design and Test of Computers, 2011

70

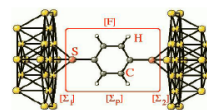
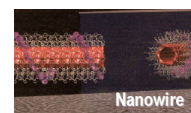
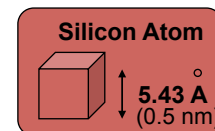
I. Integrated Circuit (IC) Technologies

1. MOS Technology
 1. The MOSFET Transistor
 2. Transistor Performance Models
 3. MOS Technologies (nMOS, pMOS, CMOS)
2. (Bipolar Technology)
3. IC Fabrication
 1. Fabrication Process
 2. Example of a Diode and a MOSFET
4. Silicon Technological Evolutions
 1. Processor and Technology Roadmaps
 2. Technology Scaling
5. Interconnects

71

Silicon Technology

- 0.35 μm in 1995, 0.25 μm in 1998, 0.18 μm in 2000
- 130 nm in 2002, 90 nm in 2004, 65 nm in 2007
- 45 nm in 2010 (first chip 2008)
 - 11-15 metal levels, wafer 30cm
 - 0.6-0.9 Volts
 - 700 MHz (ASIC) - 9 GHz (on-chip 12 inverters) - 5 GHz (off-chip)
 - 3-4 (MPU), 1 (DRAM) - 4-8 (ASIC) cm²
 - DRAM: 4Gbits, 4Gbits/cm², 0.005 \$/Mbits
 - 300 (MPU) - 6000 (ASIC) MTr/cm², 0.05-0.1 \$/MTr (MPU)
 - SRAM: 1500MTr/cm², 250Mbits/cm²
 - 6000 RISC processors (e.g. ARM7)
- 28 nm in 2013 (first chip in 2010)
- 11 nm in 2019-2021 and then ?
- Post-Silicon Technologies (nanotechnologies)



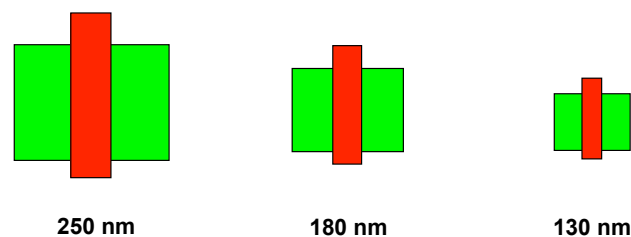
Silicon in 2015

- Power Supply: 0.6-0.8 V
- Technology: 25 nm CMOS (200 Ang.)
 - 20 GTransistors, wafer 45 cm, 2-4 cm², 13-17 metal levels
 - Inverter 2.5 ps, 0.6 Volt
 - 33 GHz (on-chip 12 inverters) - 29 GHz (off-chip)
 - DRAM 16 GBits at 10ns, 0.006 \$/Mbits
 - SRAM (cache) 1 GBits at 1.5ns
 - 256-bit Bus
- More than 8500 Person.Month Design Cycle
- Software
- Mask set is few Millions US\$

73

Technology Scaling

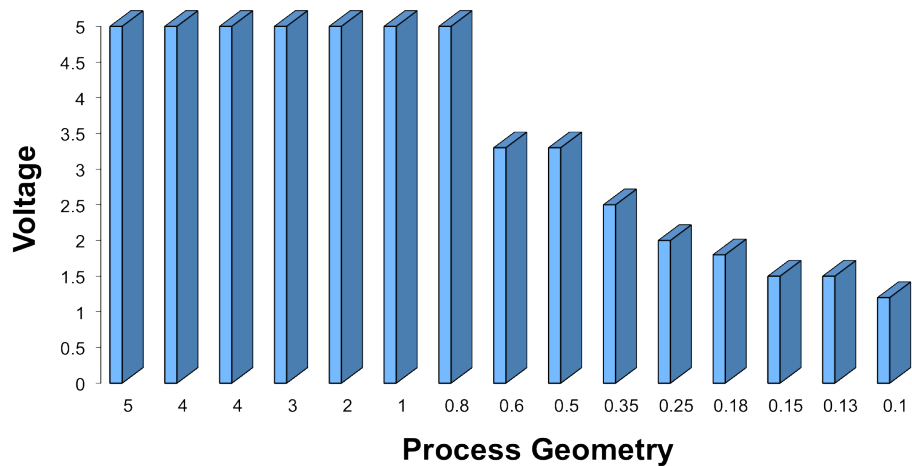
- Scaling factor: s
- Between two successive generations: $s \approx 0.7$



74

Technology Scaling

- Supply Voltage Scaling (V_{dd})



75

Technology Scaling

- Number of transistors
 - Logic: x2 every 3 years
 - Memory: x4 every 3 years
- Speed (before power wall...)
 - Logic: x2 every 3 years
 - Memory: x4 every 10 years
- Processor Performance
 - 50% per year
- Transistor density doubles and chip area increases by 25% at each new generation

76

Technology Scaling

- Power Density
 - Frequency is increased by 43%
 - Total capacitance and supply voltage (Vdd) are decreased by 30%
 - Then **Energy** is decreased by 65%
 - $E = C.V_{dd}^2 = 0.7C' \times (0.7V_{dd}')^2 = 0.35C'.V_{dd}'^2 = 35\% E'$
 - **Power** is decreased by 50%
 - $P = f.C.V_{dd}^2 = 1.43f' \times 0.35C' \times V_{dd}'^2 = 50\% P'$
 - Activity is supposed constant
 - But this is for a constant transistor count!
 - **Power Density** therefore increases by 2
 - Power supply current increases a lot
 - 100W at 1v equals to...?

77

Technology Scaling

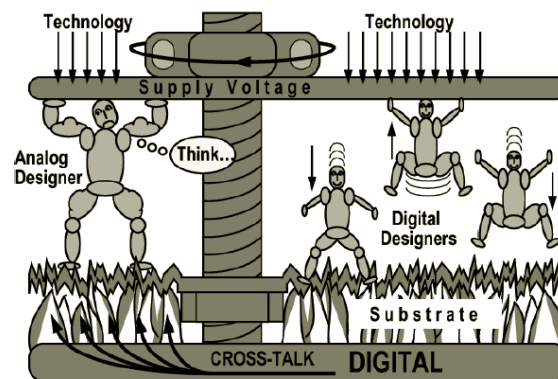
- Scaling factor: s
- Between two successive generations: $s \approx 0.7$

Device dimensions : W, L, tox, junction depth	s
Transistor area (W.L)	s^2
Capacitance per unit area : C_{ox}	$1/s$
Capacitances : $C=WLC_{ox}$	s
Vdd, Vt	s
Gate delay	s
Power/gate	s^2
Power.delay product	s^3
Power density	1

78

Power Supply Voltage Evolution

- Power and Substrate Noises
 - Vdd scaling → SNR ↘



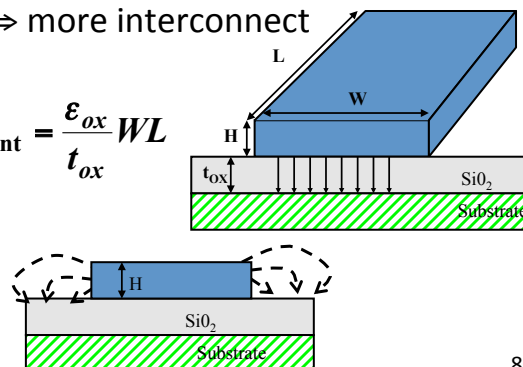
[© R. Rutenbar, CMU]

79

1.5 Interconnections

- Why worry about interconnections?
 - Important gate load
 - Technology scaling ⇒ more parasitic effects
 - Increase of chip area ⇒ more interconnect

- Capacitance model: $C_{\text{int}} = \frac{\epsilon_{\text{ox}}}{t_{\text{ox}}} WL$
- Side capacitance
- Crosstalk capacitance

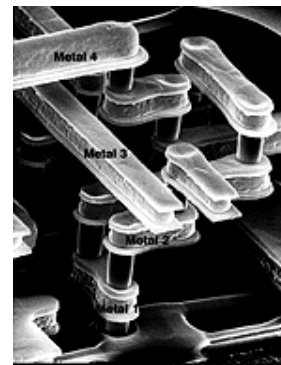
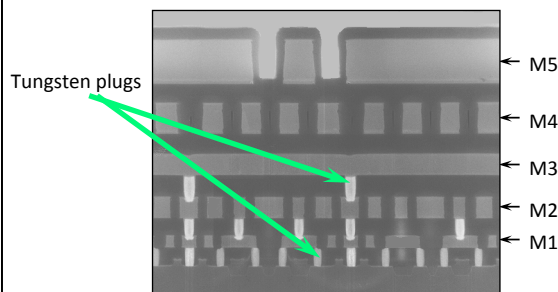


80

Interconnections

- Resistance model
 - With technology scaling, parasitic resistance grows in $1/H$
 - Solutions:
 - Keep H constant
 - Increase number of wiring levels
 - Optimised contact resistance

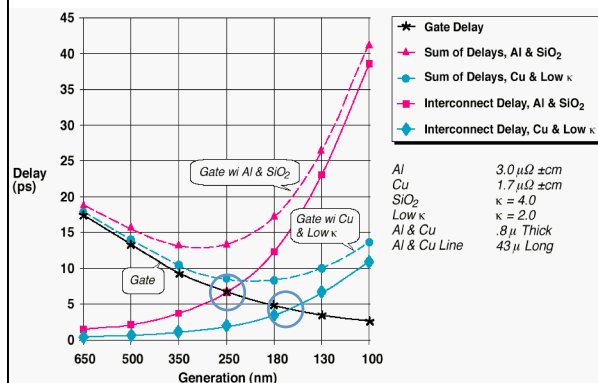
$$R = \rho \frac{L}{HW}$$



81

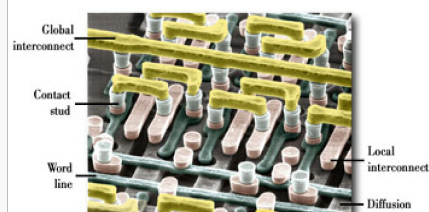
Interconnections in Deep Submicron

- Delay is fundamentally changed
 - Wire interconnections dominate delay and power
 - 60% of delay critical path is due to wires
 - Wire routing prediction is necessary at gate level
- Use of copper, increase metal levels

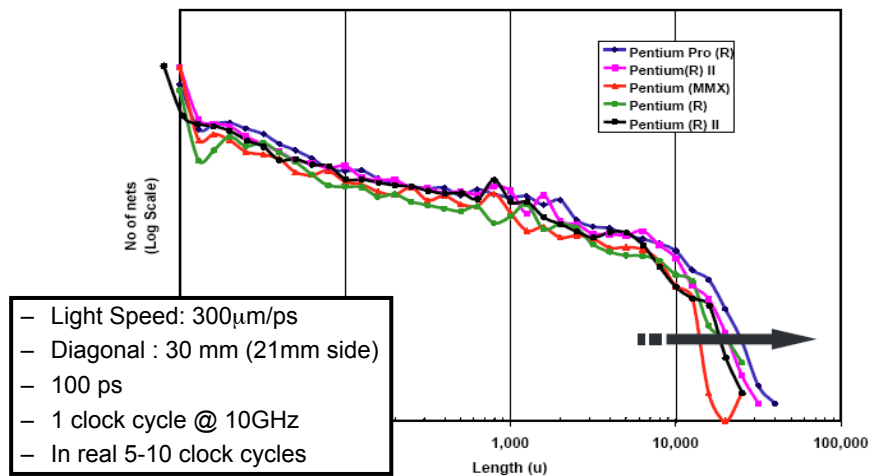


Example: delay of a 2-input NAND

- connected to 2mm of metal: 280ps
- connected to 0.5mm of metal: 119ps



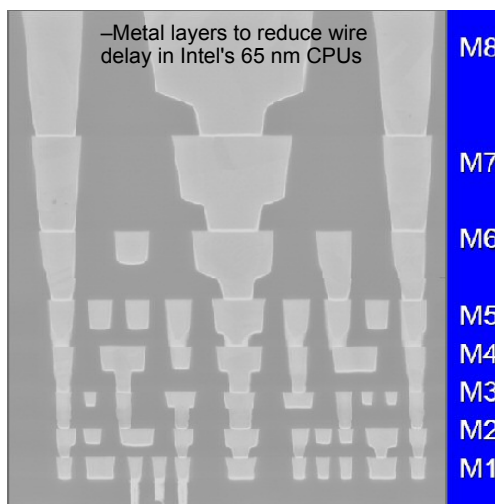
Interconnection Length



[Source: Intel]

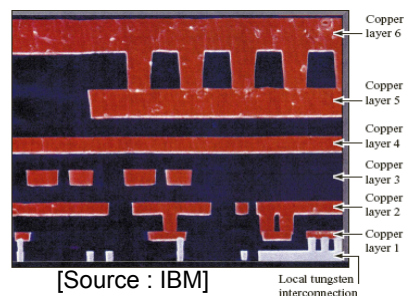
83

Reducing wire delay



[Source: Intel]

- Height of wires
- Copper
- Repeaters

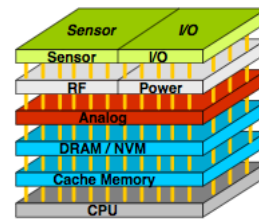


[Source : IBM]

84

Circuits go 3D

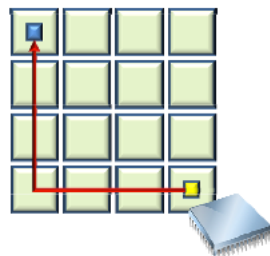
- 3D Integrated Circuits
- Stack Multiple Dies
- Connect Dies with Through Silicon Vias (TSV)
- Examples
 - Image Sensors
 - Processor + Memory



85

Why 3D ?

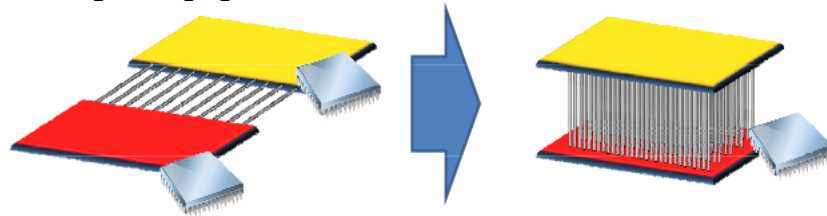
- Wire Length Reduction
 - Replace long, high capacitance wires by TSVs
 - Low Latency, Low Energy
- Small footprint



86

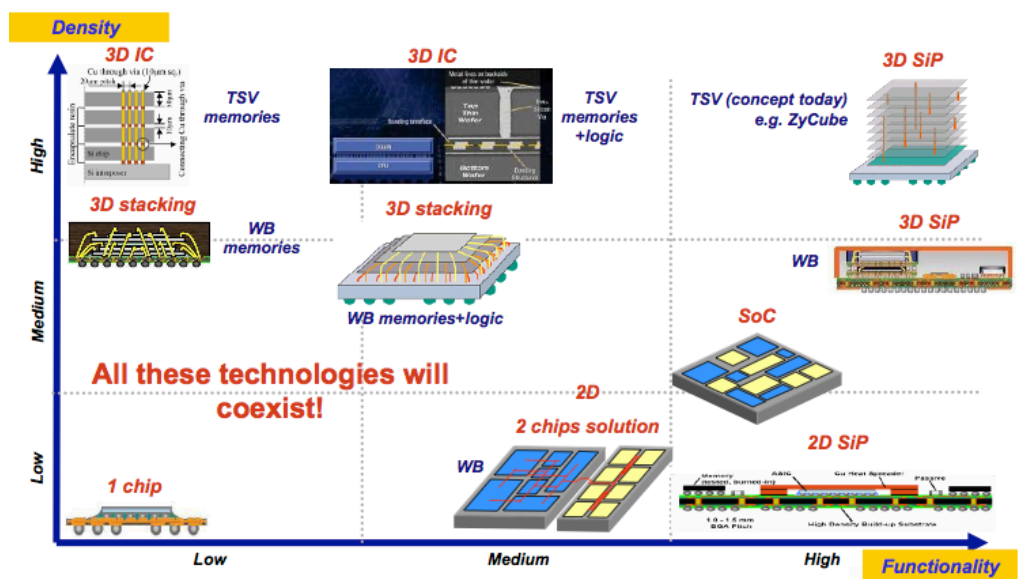
Why 3D ?

- Integration
 - From “Off-Chip” to “On-Chip”
 - Improved Communication
 - Low Latency, High Bandwidth, and Low Energy
 - Heterogeneous Integration
 - E.g. Emerging Devices



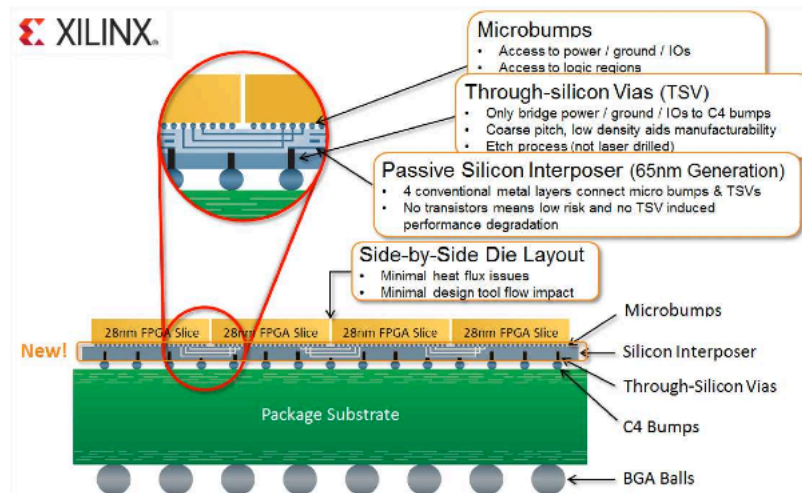
87

3D Technologies



Circuits go 2.5D...

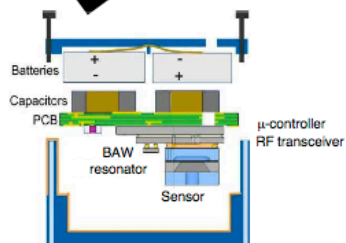
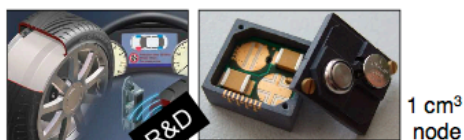
–Smaller chip size = increase yield!



89

Circuits go 3D

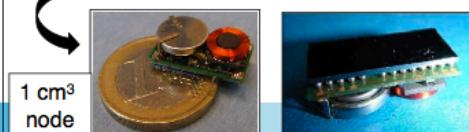
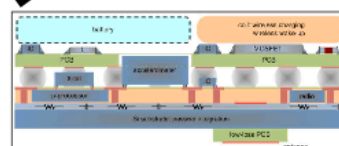
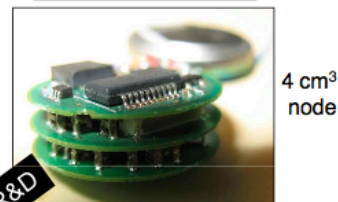
Tire Pressure Monitoring



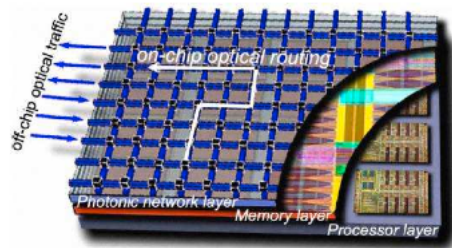
Target: batteryless → Vibration

STMicroelectronics

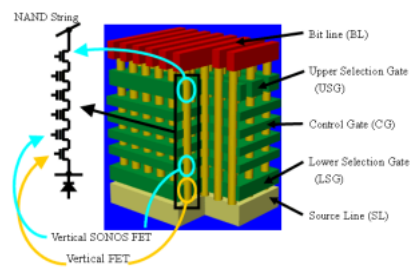
Health & Fitness



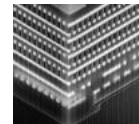
Circuits go 3D



Source: IBM



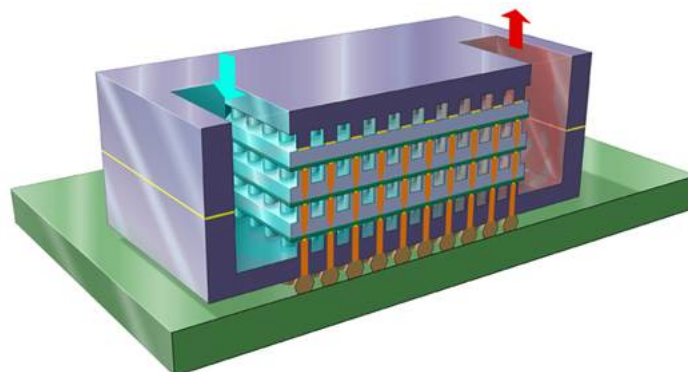
Toshiba NAND
Flash Memory



91

Cooling!

- Thermal effects



92

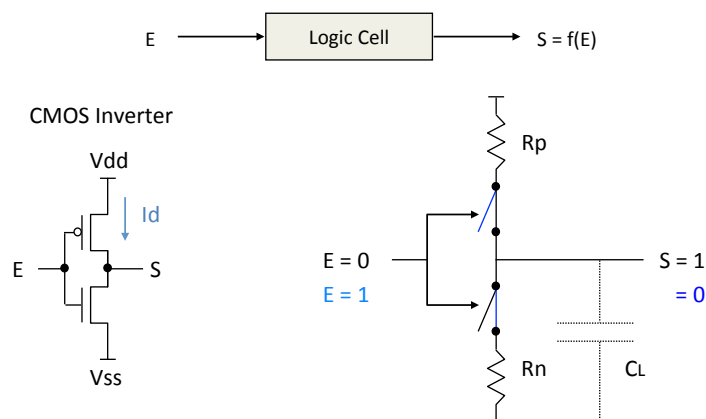
II. Design of CMOS Cells

1. Combinatorial Logic Cells
 1. Complementary Logic (CMOS)
 2. (NMOS/Pseudo-NMOS Logic)
 3. Switch-Based Logic
2. Layout Design
 1. Design Rules
 2. Layout Design
3. Sequential Logic Cells
 1. Elementary Memory Cells
 2. ROM and RAM Memory
4. Delay and Power
 1. Delay Models
 2. Fan-In and Fan-Out
 3. Static and Dynamic Power

93

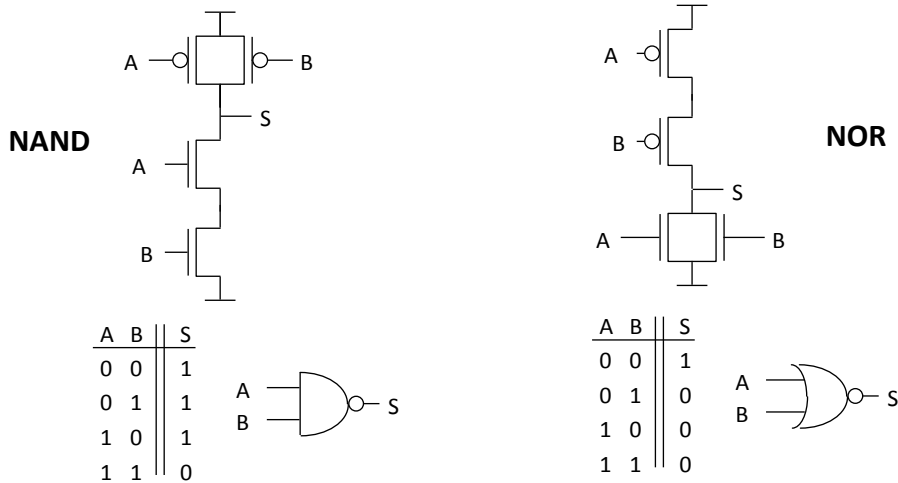
II.1 Combinatorial Logic Cells

- 1.1 Complementary Logic (CMOS)
 - CMOS Static Logic



94

NAND and NOR



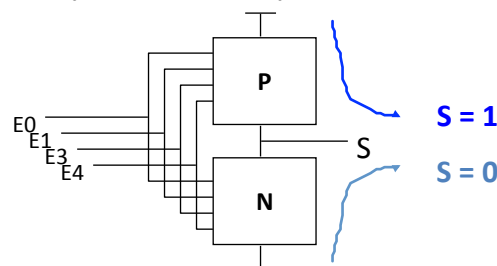
N-input Gates possible, but with Fan-In issues

95

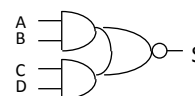
Complex gates

- One CMOS stage can generate any sum-of-product or product-of-sum:

$$S = f(E_1, E_2, \dots, E_N) = \overline{\Sigma [\Pi]} = \overline{\Pi [\Sigma]}$$

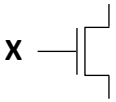
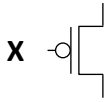
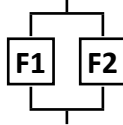
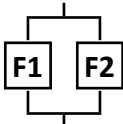


- Example: $S = \overline{A \cdot B + C \cdot D}$
AOI (And-Or-Invert) gate



96

General rules for constructing $F(X)$

F	N network	P network
X		
$F1.F2$		
$F1+F2$		

97

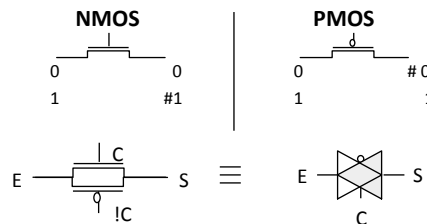
Static Logic

- Examples
 - Direct application of the design rules
 - $S1 = \overline{(A.B.D+CE+CD)}$
 - $S2 = \overline{[A.D.B+C(E+D)]}$
- Multiple-Stage Complex Functions
 - Optimisation of the logic equation
 - Trade-off between speed and area
 - $S3 = A.B.C.D$
 - $S4 = \overline{A}.B+A.\overline{B}$ (XOR)

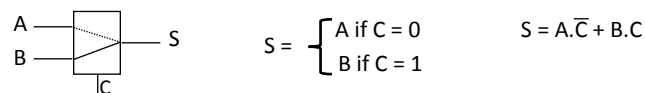
98

II.2 Pass-Transistor Logic

- Switch or Transmission Gate



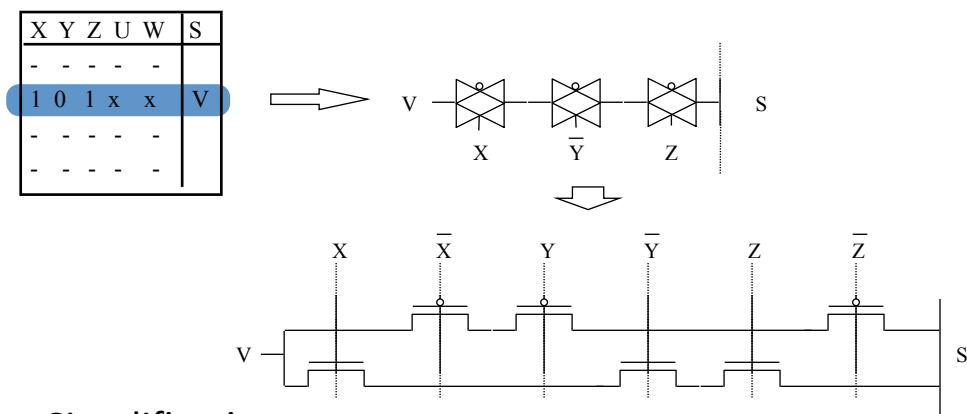
- Example: 2-input multiplexer



- Example: XOR

99

General Rules

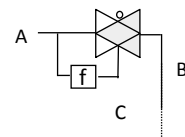
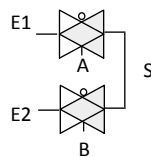


- Simplifications
 - if $V = 0$ PMOS can be omitted
 - if $V = 1$ NMOS can be omitted

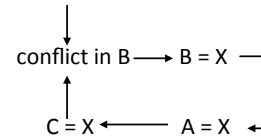
100

General Rules

- Avoid numerous serial transistors
 - voltage loss, propagation time
- Level restoring or Pull-up
 - e.g. XOR
- Avoid conflicts



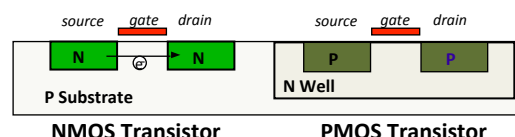
Temporary conflict when $C \Rightarrow 1$



101

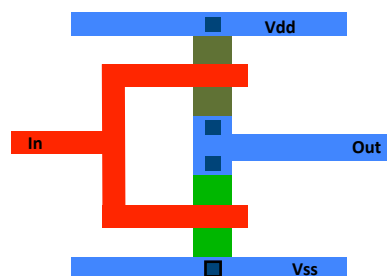
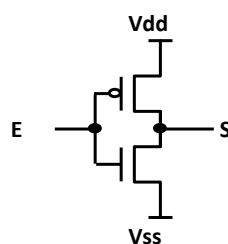
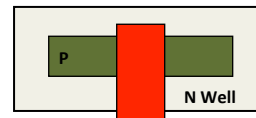
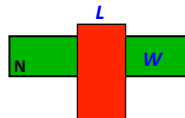
II.2 MOS Layout Design

- X nm Technology
 - X nm $\# 2\lambda$
 - λ : smallest mask size
 - Transistor Channel $L \# 2\lambda$
- Stick Diagram



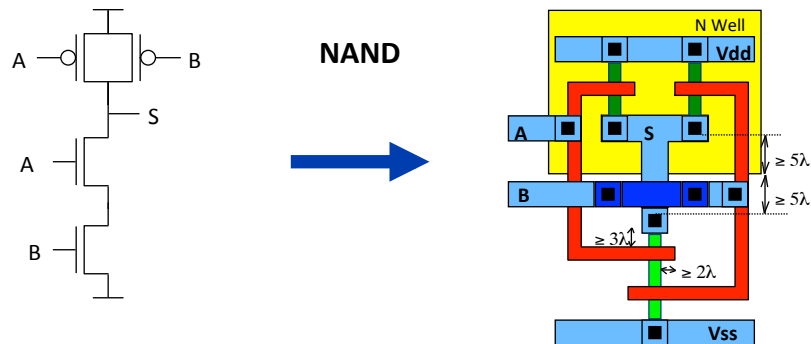
NMOS Transistor

PMOS Transistor



102

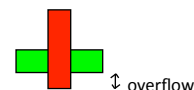
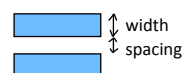
II.2 MOS Layout Design



103

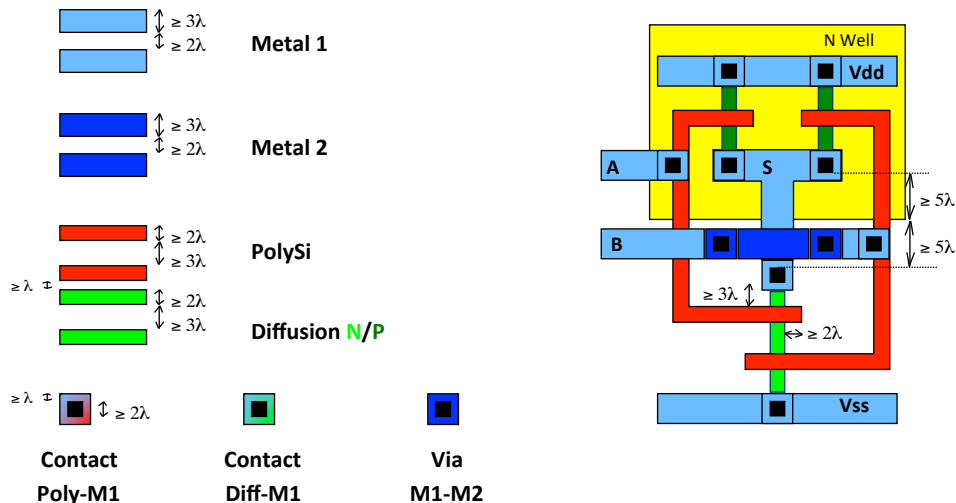
Layout Design Rules

- Interface between designer and process engineer
- Guidelines for constructing process masks
- Unit dimension: e.g. min. width
 - scalable design rules: λ
 - absolute dimensions (microns)
- Intra-Layer Design Rules
 - width: min. of each layer (poly, metal, diffusion)
 - spacing: min. space between two materials
- Inter-Layer Design Rules
 - Via, Contact, Overflow
- Layer = Color
 - Well (P,N): Yellow
 - Diffusion (P+,N+): Green
 - Polysilicon: Red
 - Metal (1,2): Blue
 - Contact to Poly: Black
 - Contact to Diff.: Black
 - Via: Black



104

Layout Design Rules: Example



105

Layer Usage

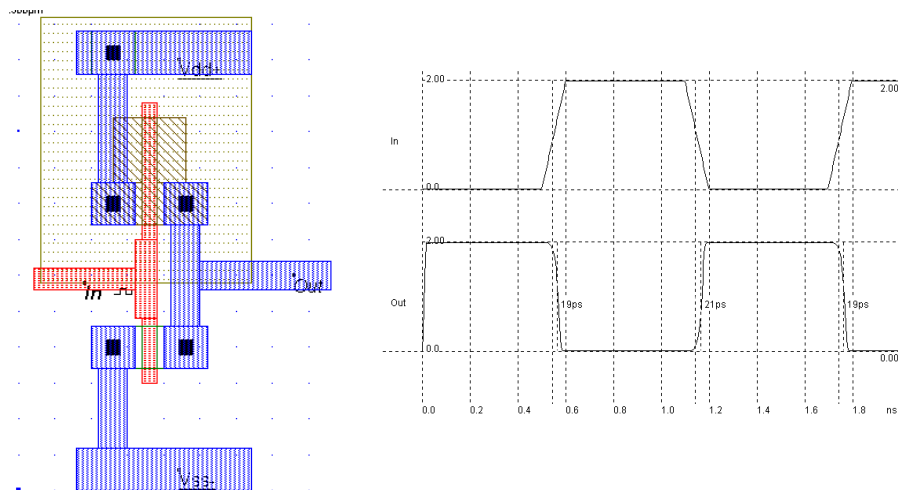
- Possible Interconnection between Layers
 - Metal1 - Poly
 - Metal1 - Diffusion P
 - Metal1 - Diffusion N
 - Metal1 - Metal2 (via)
- Metal 1 and 2: low resistivity
 - Long Interconnect
 - Vdd, Vss
- Polysilicon: higher resistivity
 - Middle-length Interconnect, resistor
 - e.g. intre-cell gate interconnections
- Diffusion: very-high resistivity
 - Source/Drain short interconnections
 - High Capacitance

106

Example: inverter

demo

- Microelectronics practical work (TP) on MicroWind



107

II. Design of CMOS Cells

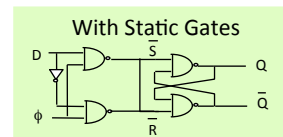
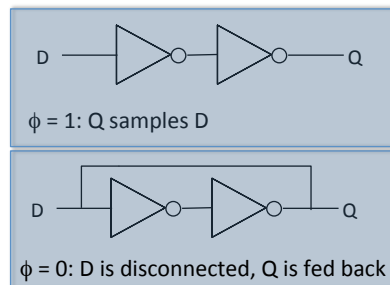
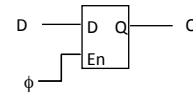
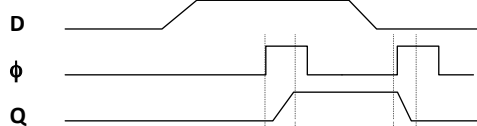
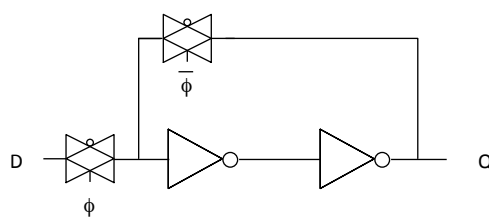
1. Combinatorial Logic Cells
 1. Complementary Logic (CMOS)
 2. (NMOS/Pseudo-NMOS Logic)
 3. Switch-Based Logic
2. Layout Design
 1. Design Rules
 2. Layout Design
3. Sequential Logic Cells
 1. Elementary Memory Cells
 2. ROM and RAM Memory
4. Delay and Power
 1. Delay Models
 2. Fan-In and Fan-Out
 3. Static and Dynamic Power
5. Interconnections

108

II.3 Sequential Logic Circuits

3.1 Elementary Memory Cells

1. Static Memory Basic Cell: Latch

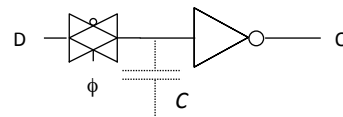


109

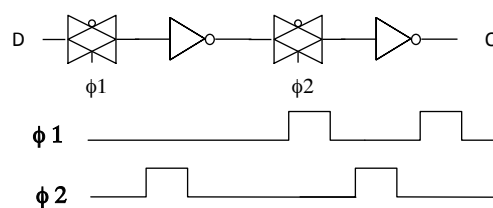
II.3 Sequential Logic Circuits

2. Dynamic Memory Cell

- MOS Capacitor: $C = f(\text{area})$
- State ('1') (voltage level) is stored for few ms
 - Leakage Current
 - Need for refreshing state



Ex. Shift Register

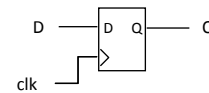
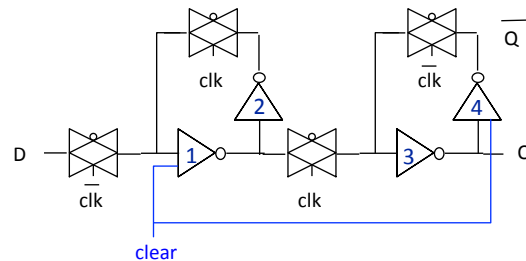


110

II.3 Sequential Logic Circuits

• 3. D Flip-Flop (edge-triggered)

- Two serial latches



- D is sampled in inverter (1) when $\text{clk} = 0$
- Latch (1) and (2) keeps D value when $\text{clk} = 1$ until !D is transferred to second latch (3) and (4)
- Asynchronous clear signal: replace inv. (1) and (4) by NAND

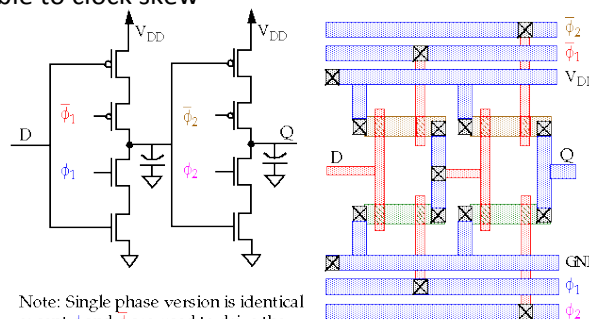
111

II.3 Sequential Logic Circuits

• D flip-flop optimisations

- C²MOS Flip-Flop – negative edge triggered

- Higher speed
- Less sensible to clock skew



Note: Single phase version is identical except ϕ and $\bar{\phi}$ are used to drive the n/p-trans on the left while $\bar{\phi}$ and ϕ are used to drive the n/p-trans on the right.

112

II.3.4 ROM and RAM Memory

• Memory Classification

Random-Access Read-Write Memory	Non-Random Access Read-Write Memory	Non-Volatile Read-Write Memory	Read-Only Memory
SRAM DRAM	FIFO LIFO Shift-Register CAM	EPROM EEPROM Flash	Mask-Programmed ROM Programmable ROM

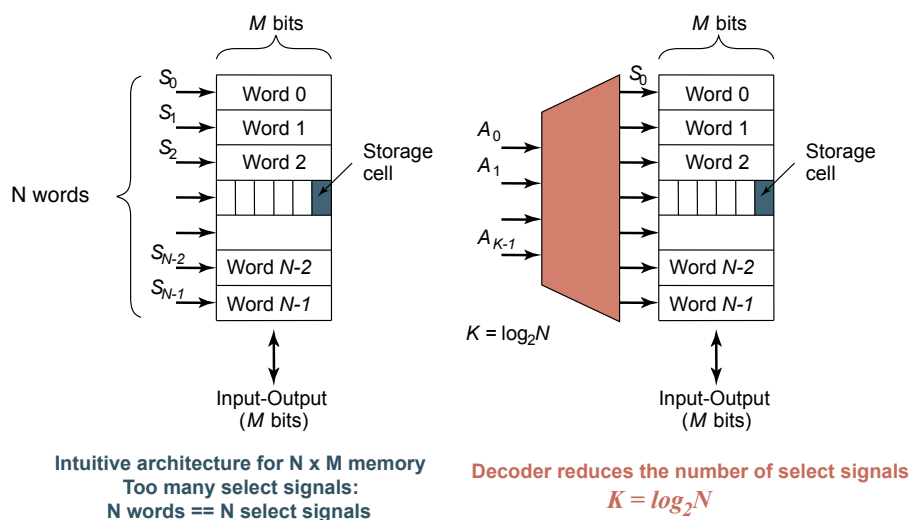
• Parameters

– Address Size, Word Size, Access Time R/W, ...

Adapted from J. Rabaey et al., Digital Integrated Circuits, Second Edition. Copyright 2003 Prentice Hall/Pearson.

113

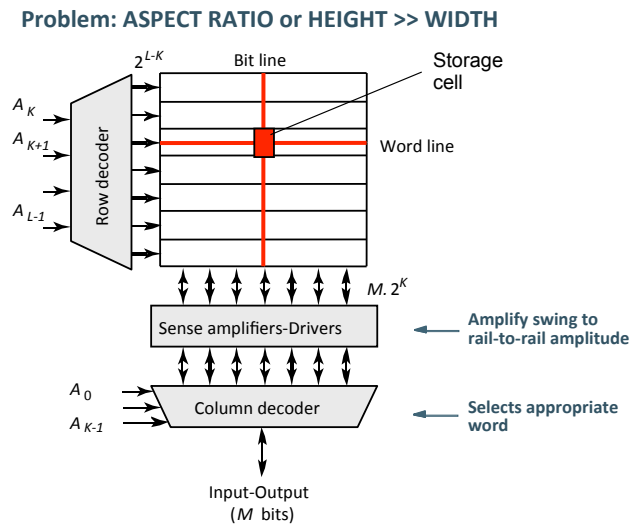
Memory Architecture



Adapted from J. Rabaey et al., Digital Integrated Circuits, Second Edition. Copyright 2003 Prentice Hall/Pearson.

114

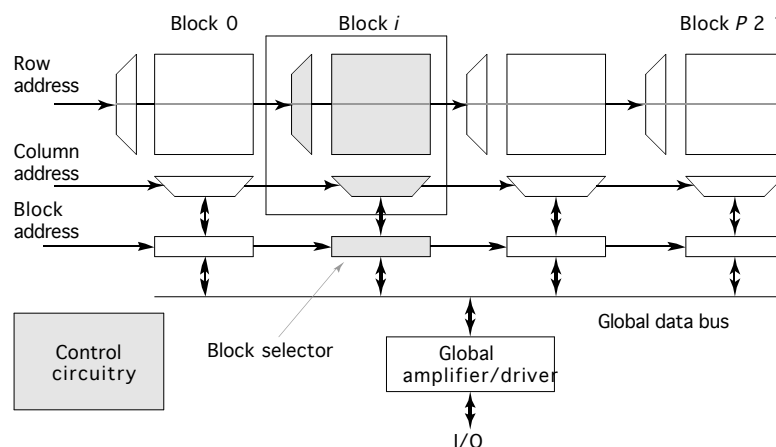
Memory Architecture



Adapted from J. Rabaey et al., Digital Integrated Circuits, Second Edition. Copyright 2003 Prentice Hall/Pearson.

115

Memory Architecture: Hierarchy



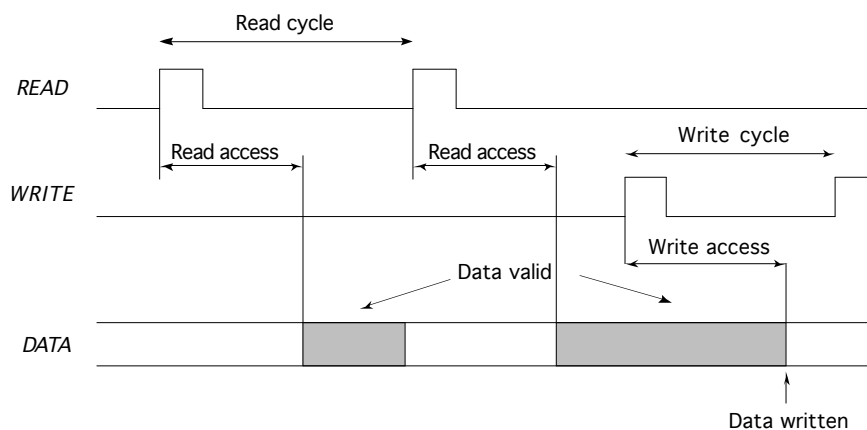
Advantages:

1. Shorter wires within blocks
2. Block address activates only 1 block => power savings

Adapted from J. Rabaey et al., Digital Integrated Circuits, Second Edition. Copyright 2003 Prentice Hall/Pearson.

116

Memory Timing



Adapted from J. Rabaey et al., Digital Integrated Circuits, Second Edition, Copyright 2003 Prentice Hall/Pearson.

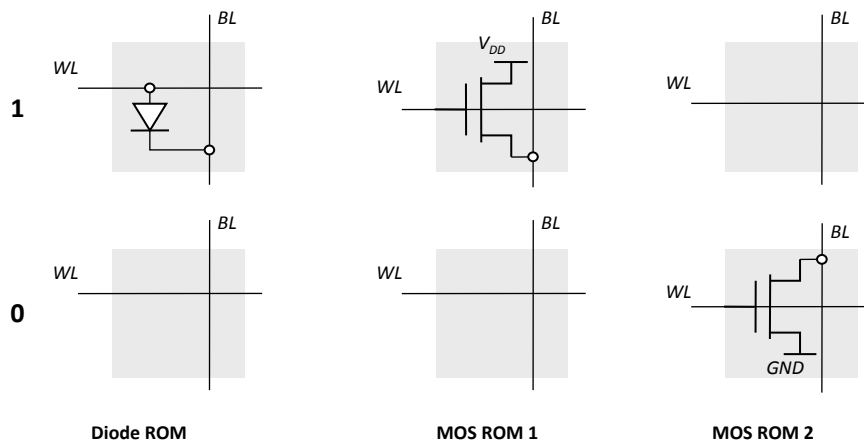
117

Elementary Memory Cell

- Non-volatile memory
 - ROM: diode, mask programmed
 - PROM: transistor, one-time programmable (metallization, avalanche effect)
 - EPROM: floating-gate MOS, programmable (avalanche), erasable by UV
 - Flash, EEPROM: floating-gate MOS
 - avalanche injection of electrons in the floating gate results in higher V_t , erasable electrically (tunnel effect)
- Read-write memory
 - DRAM: dynamic latch (capacitance)
 - simple cells, refreshing cycle (# ms)
 - SRAM: static latch
 - complex cells

118

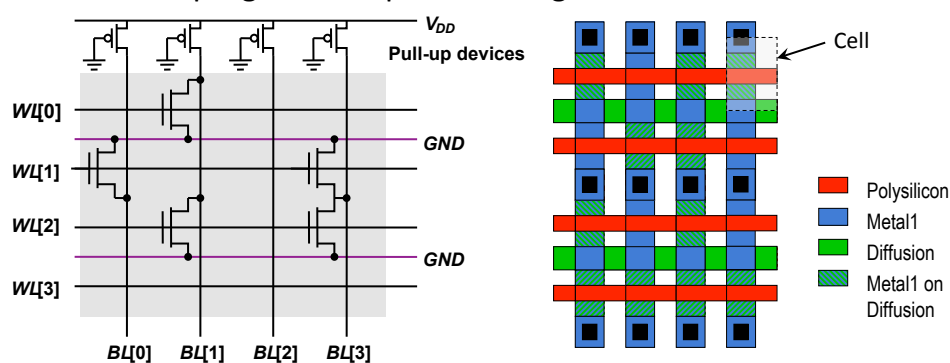
Read-Only Memory Cells



119

ROM Memory (NOR type)

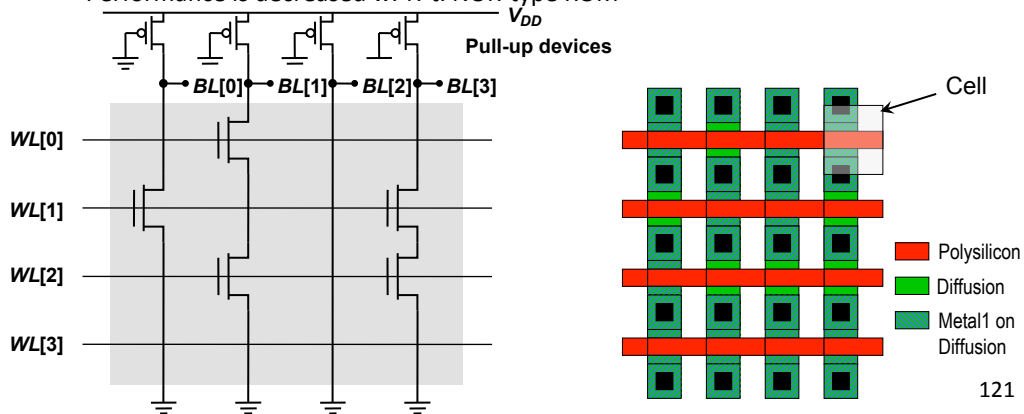
- Memory Cell $P[i,j]$: row $WL[i]$ (word line) and column $BL[j]$ (bit line) selected
 - With a transistor: $WL[i]=1$, transistor is ON, $BL[j]=0$
 - Without a transistor: $BL[j]=1$
 - Mask programmed par including or not transistors



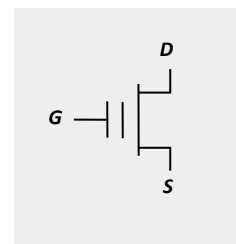
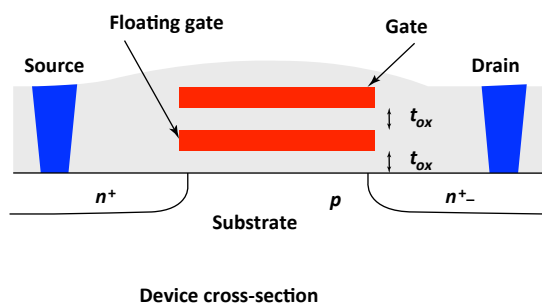
120

ROM Memory (NAND type)

- All word lines high ($WL[j]=1$) by default with exception of selected row $WL[i]=0$
- Without a transistor: $BL[j]=1$
- With a transistor: $WL[i]=0$, transistor is ON, $BL[j]=0$
- No V_{dd} or V_{ss} connections: size is decreased
- Performance is decreased w.-r.-t. NOR-type ROM



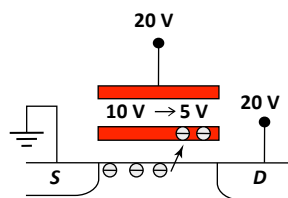
Floating-gate transistor



Schematic symbol

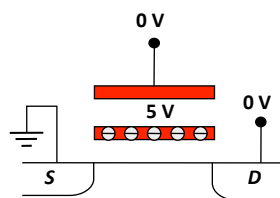
122

Floating-Gate Transistor Programming



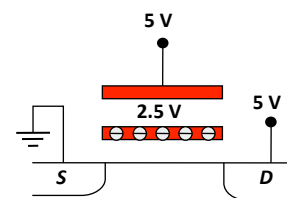
Avalanche injection

High voltage creates an Avalanche Effect.
Electrons are trapped on the floating-gate



Removing programming voltage leaves charge trapped

Electrons stay trapped for a lower voltage



Programming results in higher V_T

Applying $V_{dd}=5V$, transistor effect will not happen since threshold voltage V_t is now 7.5V

123

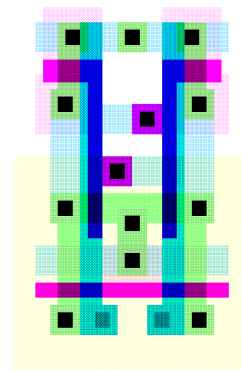
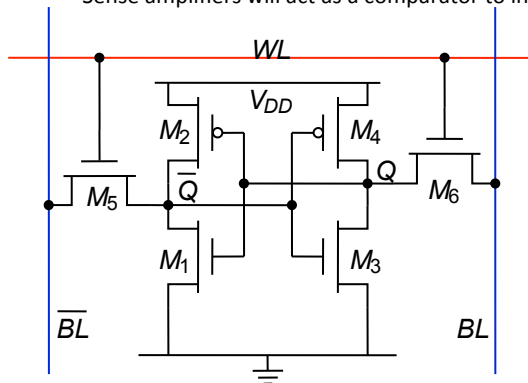
Read-Write Memories (RAM)

- Static RAM (SRAM)
 - Data stored as long as supply is applied
 - Large (6 transistors/cell)
 - Fast
 - Differential
- Dynamic RAM (DRAM)
 - Periodic refresh required
 - Small (1-3 transistors/cell)
 - Slower
 - Single Ended

124

6-Transistor CMOS SRAM Cell

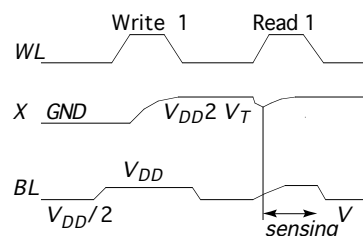
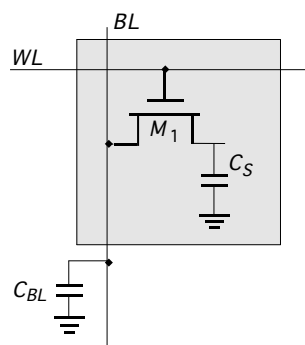
- Latches where WL replaces clock
- Dual-rail bit-lines required to increase noise margin during R/W
- WL selection: $WL[i] = 1$
- Write 0: $BL=0$ et $!BL=1 \Leftrightarrow$ Reset of Latch
- Read: BL et $!BL$ pre-charged to 1, WL selection $\rightarrow BL=Q$ and $!BL=!Q$
 - Sense amplifiers will act as a comparator to increase speed of Latch value to output



125

1-Transistor DRAM

- Refresh: read followed by write (every 2-4ms)
- Write: $WL = 1$, input on BL , C_s is charging/discharging
- Read: $WL = 1$, BL pre-charged to $V_{DD}/2$, charge exchange between C_s and C_{bl} , read is destructive (need for refresh)
- Amplification is necessary after read
- C_s is a junction capacitor (transistor)

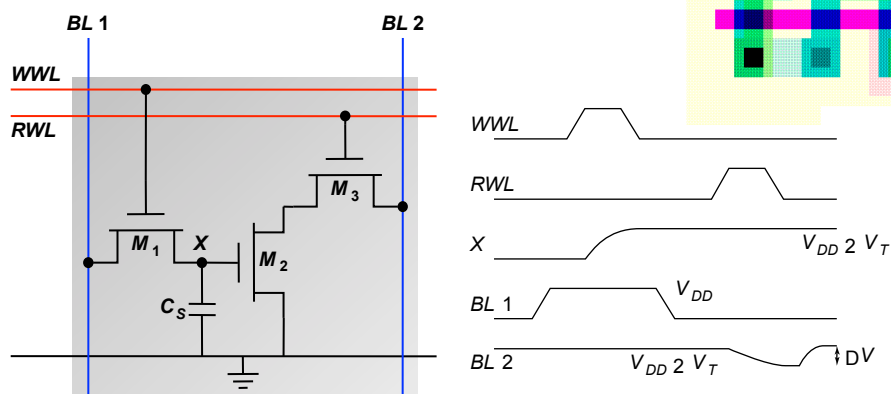


$$\Delta V = (V_{bit} - V_{DD}/2) \times \frac{C_s}{C_s + C_{BL}}$$

126

3-Transistor DRAM

- 2 lines WL and BL: read and write
- No amplification



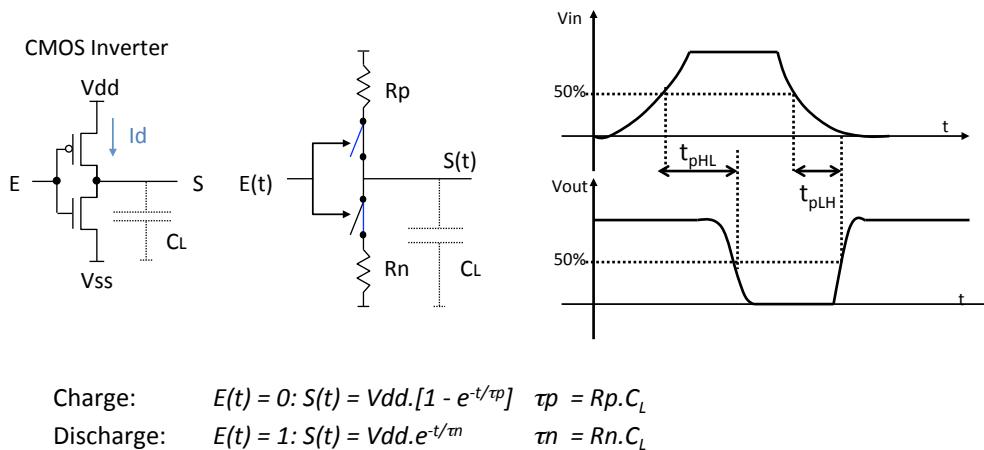
127

II. Design of CMOS Cells

1. Combinatorial Logic Cells
 1. Complementary Logic (CMOS)
 2. (NMOS/Pseudo-NMOS Logic)
 3. Switch-Based Logic
2. Layout Design
 1. Design Rules
 2. Layout Design
3. Sequential Logic Cells
 1. Elementary Memory Cells
 2. ROM and RAM Memory
4. Delay and Power
 1. Delay Models
 2. Fan-In and Fan-Out
 3. Static and Dynamic Power
5. Interconnections

128

II.4.1 Propagation Time

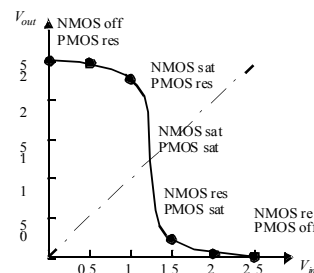
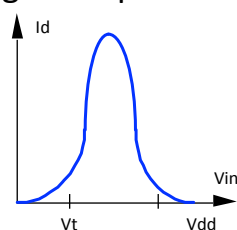


$$T_p = \text{MAX}(T_{p_{LH}}, T_{p_{HL}})$$

129

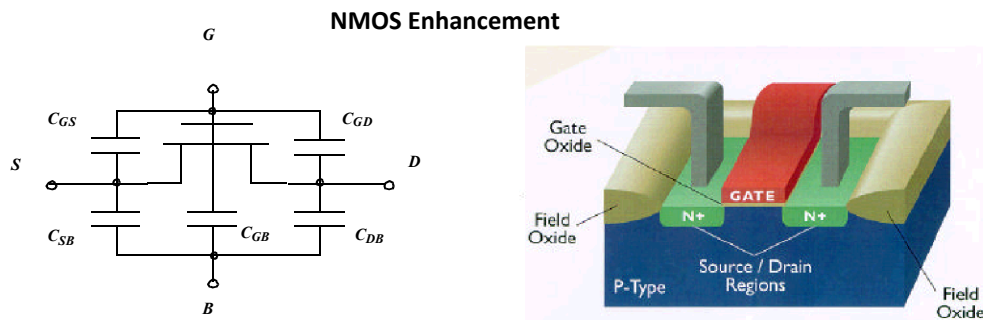
I/O Transfer Characteristic

- Stable states: $E = V_{ss}$ or V_{dd}
 - Static power $\neq 0$ (only transistor leakage)
- Transition state: $V_t < E < V_{dd} - V_t$
 - NMOS and PMOS partially opened
 - Short circuit power: $P = F_T \cdot C_{eff} \cdot V_{dd}^2$, $C_{eff} = C_L \cdot \text{Prob}(0 \leftrightarrow 1)$
 - Signal slopes have to be fast



130

NMOS Parasistic Elements



$$\text{Drain-Source Resistance: } R_{DS} = \frac{1}{K(V_{dd} - V_t)} = \frac{1}{k(V_{dd} - V_t)} \frac{L}{W}$$

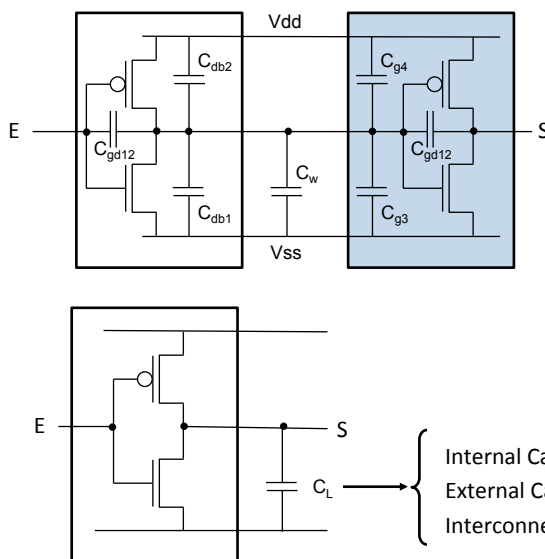
$$\text{Gate Capacitor: } C_g = \frac{\epsilon \cdot W \cdot L}{t_{ox}} = W \cdot L \cdot C_{ox}$$

$$\text{Drain/Source-Bulk Capacitor: } C_{sb} = C_{db} \approx W \cdot L \cdot C_j$$

$$\text{Delay: } \tau = R_{DS} C_g = \frac{L^2}{\mu(V_{dd} - V_t)}$$

131

Gate/drain/source/bulk capacitances



C_{db2} : Pmos drain capa. (C_{DP})

C_{db1} : Nmos drain capa. (C_{DN})

C_{gd12} : Nmos/Pmos gate-drain capa.

$C_{DP} + C_{DN} = C_{DN}(1+a)$

with $a = (W/L)_P / (W/L)_N$

$C_{int} = C_{db1} + C_{db2} + 2 \cdot C_{gd12}$

C_w : wire capa. = C_{itx}

C_{g3}, C_{g4} : gate capa.

$C_{ext} = C_{g3} + C_{g4}$

$C_L = C_{int} + C_{itx} + C_{ext}$

Internal Capacitance: C_{int}
External Capacitance: C_{ext}
Interconnect Capacitance: C_{itx}

132

Inverter Propagation Time

$$t_{nHL} \approx \frac{1}{k_n} \frac{C_L}{(V_{DD} - |V_{Tn}|)} = \frac{L^2}{\mu_n \cdot (V_{DD} - |V_{Tn}|)}$$

$$t_{pLH} \approx \frac{1}{k_p} \frac{C_L}{(V_{DD} - |V_{Tp}|)} = \frac{L^2}{\mu_p \cdot (V_{DD} - |V_{Tp}|)}$$

$$t_p = \frac{t_{pLH} + t_{nHL}}{2} \approx \frac{C_L}{2(V_{DD} - |V_T|)} \left(\frac{1}{k_n} + \frac{1}{k_p} \right)$$

$$R_m = \frac{1}{2(V_{DD} - |V_T|)} \left(\frac{1}{k_n} + \frac{1}{k_p} \right)$$

$$t_p = C_L \int \frac{dv}{i_{DS}(v)} = C_L \frac{V_{DD}}{2} \overline{I_{DS}}$$

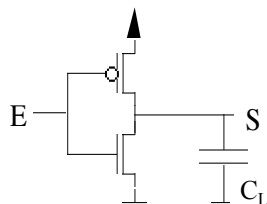
$$I_{ds} = \begin{cases} 0 & \text{Off} \\ k \left[(V_{GS} - V_t) V_{DS} - \frac{V_{DS}^2}{2} \right] & \text{Linear} \\ \frac{k}{2} (V_{GS} - V_t)^2 & \text{Saturated} \end{cases}$$

133

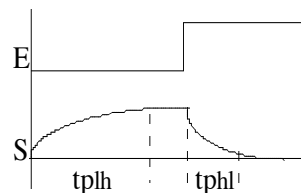
Simplified model

$$t_p = R_{DS} \cdot C_L = R_{DS} \cdot [C_{int} + C_{itx} + C_{ext}]$$

- R_{DS} is transistor drain-source resistance
- Temperature $\uparrow$: mobility $\downarrow$; propag. time $\uparrow$; I_{ds} max $\downarrow$



$$t_{plh} = R_p \cdot C_l$$



$$t_{phl} = R_n \cdot C_l$$

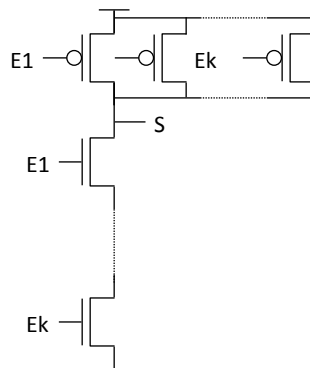
R_n : NMOS resistance

R_p : PMOS resistance ($R_p \neq 2 \cdot R_n$)

134

Exemples

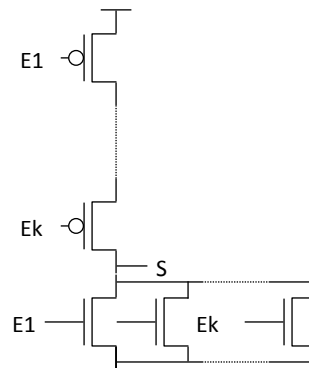
k-input NAND



$$t_{plh} =$$

$$t_{phl} =$$

k-input NOR



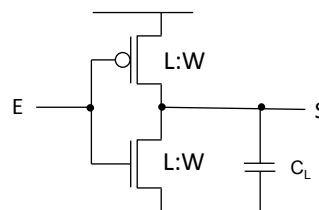
$$t_{plh} =$$

$$t_{phl} =$$

135

Transistor Sizing

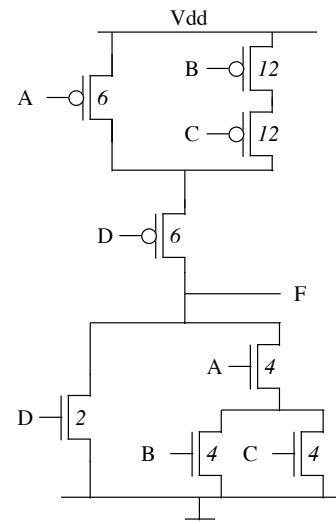
- CMOS inverter
 - A given technology gives, for a NMOS transistor with size $2\lambda:2\lambda$ (L:W), values for R_{nu} et Cg_{nu} of 1250Ω and $0.3fF$.
 - For NMOS transistors with size $2\lambda:6\lambda$, and PMOS with size $2\lambda:12\lambda$, give the values of:
 - $R_n =$ $R_p =$
 - $C_{gn} =$ $C_{gp} =$
 - What is the delay of this inverter loaded by the same inverter ?



136

Transistor Sizing

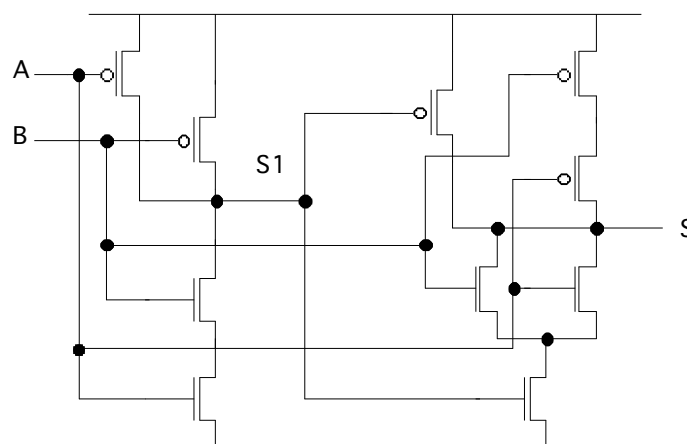
- Complex function
 - Same technology as previous inverter
 - $T_{plh} =$
 - $T_{phl} =$
 - Indicate critical path
 - Which input values give the best/worst case delay?



137

Propagation Delay of Cascaded Cells

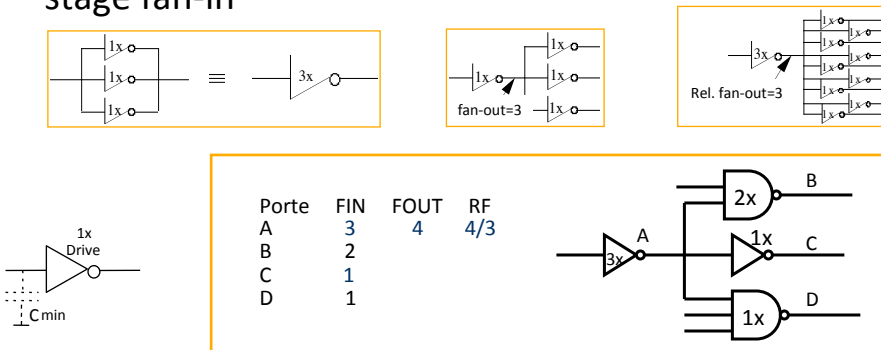
- Give $S1$ as a function of A and B .
- Give S as a function of $S1$, A and B .
- Give gate count and propagation delay of the cell.



138

II.4.2. Fan-in and Fan-out

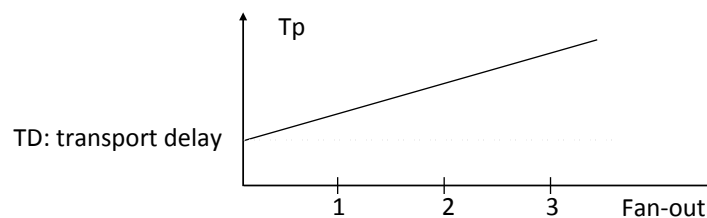
- Fan-In (or Drive): relative to size of transistors
 - Basic inverter is 1x
- Fan-Out: ratio between load capacitance and drive
- Relative Fan-Out (RF): ratio between fan-out and next-stage fan-in



139

Logic-level model

- $T_p = \text{transport delay} + \text{inertial delay} = T_D + I_D$
- $T_p = R_{DS}[C_{int} + C_{ext}]$



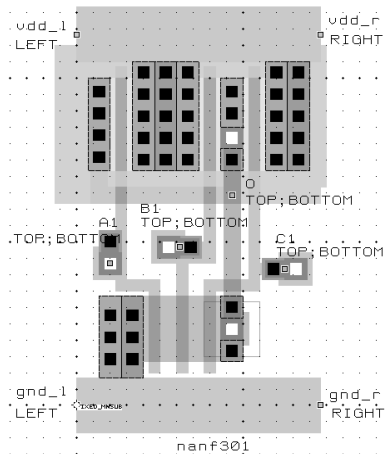
$$T_p = T_D + R_F \cdot U_D$$

U_D : unit delay

140

Cell delay

• 3-input NAND

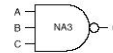


AMS **NA3** CUB 0.6 μm CMOS

NA3 is a 3-input NAND gate with 1x drive strength.

Truth Table

A	B	C	Q
L	X	X	H
X	L	X	H
X	X	L	H
H	H	H	L



Capacitance

	Cl (pF)
A	0.036
B	0.036
C	0.041

Area

0.54 mls²

Power

4.27 μW/MHz

Delay (ns) = $t_{pd} = f(SL, L)$ with: SL = Input Slope (ns); L = Output Load (pF)
Output Slope (ns) = $op_sl = f(L)$ with: L = Output Load (pF)

AC Characteristics: $T_J = 25^\circ\text{C}$, $V_{DD} = 5\text{V}$, Typical Process

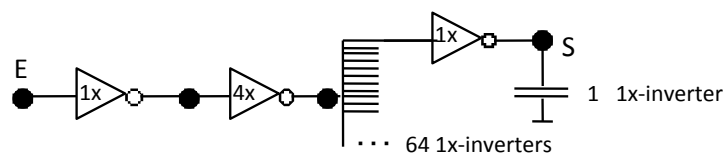
AC Characteristics

Characteristics	Symbol	SL = 0.1			SL = 2.0		
		L = 0.1	L = 0.7	L = 1.0	L = 0.1	L = 0.7	L = 1.0
Delay A to Q	t_{pdar}	0.30	1.28	1.81	0.65	1.49	1.93
	t_{pdaf}	0.24	1.03	1.49	0.13	0.91	1.27
Delay B to Q	t_{pdr}	0.28	1.25	1.78	0.60	1.45	1.90
	t_{pdrf}	0.22	1.03	1.42	0.22	1.01	1.38
Delay C to Q	t_{pdc}	0.25	1.23	1.76	0.53	1.41	1.87
	t_{pdcf}	0.20	1.02	1.40	0.29	1.12	1.50
Output Slope A to Q	op_slar	0.85	3.93	5.43	1.27	4.01	5.51
	op_slaf	0.62	3.18	4.50	1.01	3.31	4.62
Output Slope B to Q	op_slbr	0.77	3.81	5.37	1.20	3.93	5.41
	op_slbf	0.62	3.16	4.52	1.06	3.32	4.63
Output Slope C to Q	op_slcr	0.72	3.97	5.50	1.16	4.03	5.43
	op_slcf	0.63	3.18	4.47	1.10	3.42	4.53

141

Example

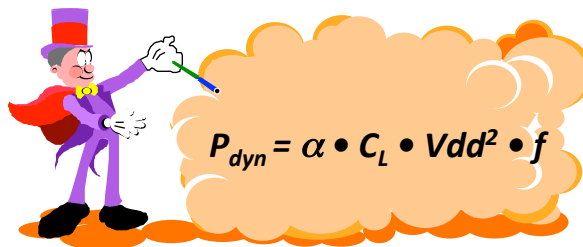
- Clock tree delay
 - Give Relative Fan-Out (RF) for all nodes
 - Estimate propagation time from E to S: T_{pES}
 - Inverter transport delay: $T_D = 0.29\text{ns}$
 - Inverter unit delay: $T_{UD} = 0.17\text{ns}$
 - Give an optimized schematic with less T_{pES}



142

II.4.3 Power

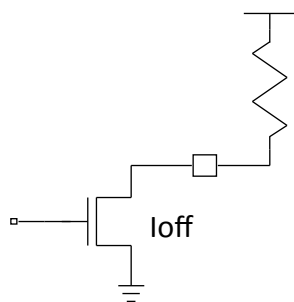
- $P = P_{dyn} + P_s = P_c + P_{sc} + P_s$
- Charging power: P_c
 - Charge and discharge of node capacitance
- Short-circuit power: P_{sc}
 - Short circuit during the commutation of logical structures
- Static power: P_s
 - Junctions, sub-threshold behaviour



143

Static power (1)

- Sub-threshold Leakage Current
 - Even if $V_{gs} < V_t$ MOS transistors are not completely off
 - If V_{dd} decreases (towards V_t) leakage currents increase quickly
- Source/Drain-Bulk junction leakage (diodes)



Negligible as $V_{dd} \gg V_t$

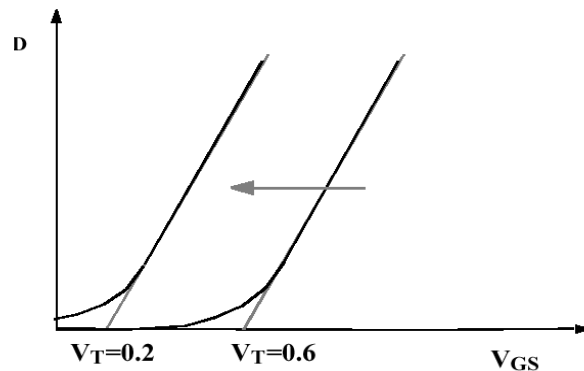
$$I_{off} = I_0 e^{-\frac{V_T}{n \cdot U_T}}$$

$$P_s = N_{Tr} \cdot V_{dd} \cdot I_0 \cdot e^{-\frac{V_T}{n \cdot U_T}}$$

144

Static power(2)

- Impact of threshold voltage



- Recent technologies use transistors with 2 V_t
 - Low-Leakage or High-Performance cells

145

Static power (3)

- 130 nm Technology

$I_{static}(A)$	slow-slow	typical	fast-fast
-10°C	2.1E-06	1.2E-05	7.0E-05
25 °C	1.7E-05	8.2E-05	3.9E-04
50 °C	6.1E-05	2.5E-04	1.1E-03

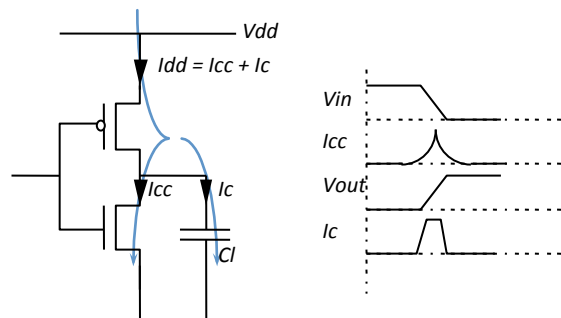
- Circuit with 5 million transistors
 - 0.6 Volt, 1 mA leakage current
 - Higher than total specified current...
- Cannot be still neglected!

[Piguet03]

146

Dynamic power (1)

- Charging current: I_c



$$P_c = \alpha \cdot f \cdot C_L \cdot V_{dd}^2$$

α : activity, C_L : total load capacitance, f : frequency

147

Dynamic power (2)

- Energy per transition = $C_L V_{dd}^2$
- Power = Energy per transition x rate of transition

$$P_c = C_L V_{dd}^2 f_{0 \rightarrow 1}$$

$$P_c = C_L V_{dd}^2 f \text{Prob}_{0 \rightarrow 1}$$

$$P_c = \alpha C_L V_{dd}^2 f$$

$$P_c = C_{EFF} V_{dd}^2 f$$

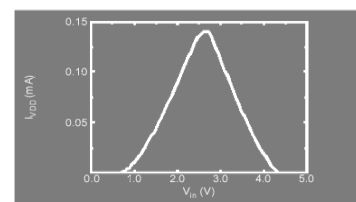
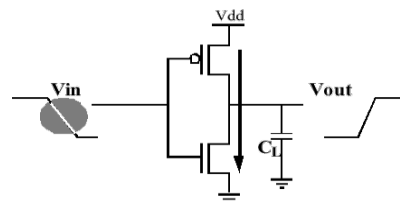
- Effective capacitance $C_{EFF} = \alpha C_L$
- Power does not directly depend on transistor size of the considered cell



148

Dynamic power (3)

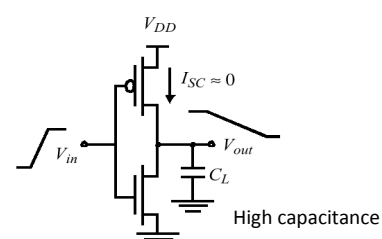
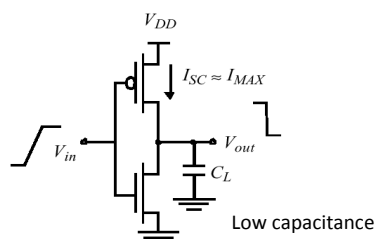
- Short-circuit current: I_{sc}
 - Short-circuit path during commutation where NMOS and PMOS are simultaneously on
 - For correctly designed circuits: $\approx 15\%$ of total power
 - Slow slopes ?



149

Dynamic power (4)

- Short-circuit current: impact of rise/fall
 - Short-circuit time
 - Depends on gate load



$$P_{cc} = \alpha \cdot f \cdot t \cdot K \cdot (W/L) \cdot (V_{dd} - 2V_t)^3 / 2$$

α : activity, t : rise/fall time, K : techno., W/L : size

150

Dynamic power is data dependant

- Example: inverter

$$P_{0 \rightarrow 1} = P(\text{OUT} = 0) \times P(\text{OUT} = 1) = 1/4$$

- Example: 2-input static NOR gate

A	B	OUT
0	0	1
0	1	0
1	0	0
1	1	0

Hypothesis: $P(A=1) = 1/2$ et $P(B=1) = 1/2$

$$P(\text{OUT}=1) = 1/4; P(\text{OUT}=0) = 1-1/4$$

$$P_{0 \rightarrow 1} = P(\text{OUT} = 0) \cdot P(\text{OUT} = 1)$$

$$= 3/4 \times 1/4 = 3/16$$

$$C_{eff} = 3/16 \cdot C_l$$

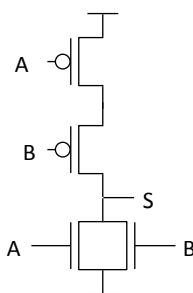
- Depends on input statistics

	$P_{0 \rightarrow 1}$
AND	$(1-P_A P_B) P_A P_B$
OR	$(1-P_A)(1-P_B) (1-(1-P_A)(1-P_B))$
XOR	$(1-(P_A+P_B-2 P_A P_B)) (P_A+P_B-2 P_A P_B)$

151

Dynamic power is data dependant

- Transition probability of a NOR gate

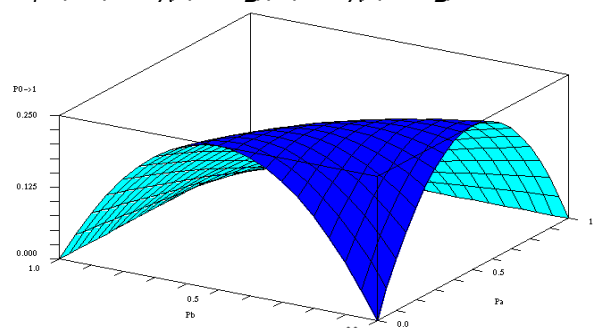


$$P_A = P(A=1); P_B = P(B=1)$$

$$P_1 = P(S=1) = (1-P_A)(1-P_B)$$

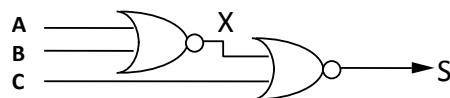
$$P_{0 \rightarrow 1} = P_0 \cdot P_1 = (1-(1-P_A)(1-P_B)) (1-P_A)(1-P_B)$$

A	B	S
0	0	1
0	1	0
1	0	0
1	1	0



Transition probabilities

- Probability propagation



$$P(A) = \frac{1}{2}$$

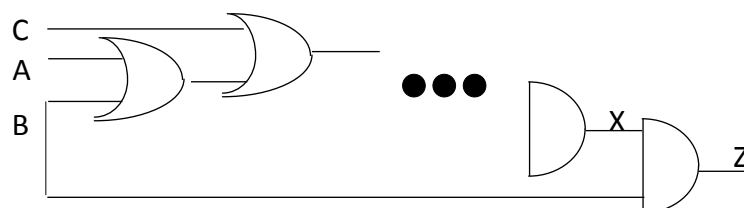
$$P(B) = \frac{1}{2}$$

$$P(C) = \frac{1}{2}$$

153

Transition probabilities

- Re-convergence: conditional probabilities

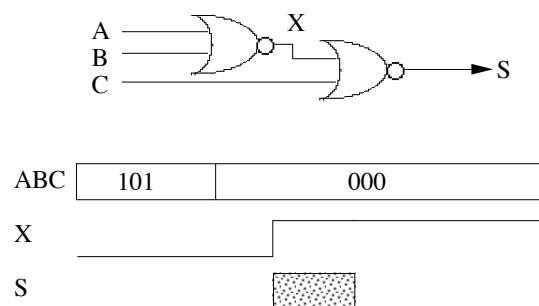


- Becomes quickly complex!

154

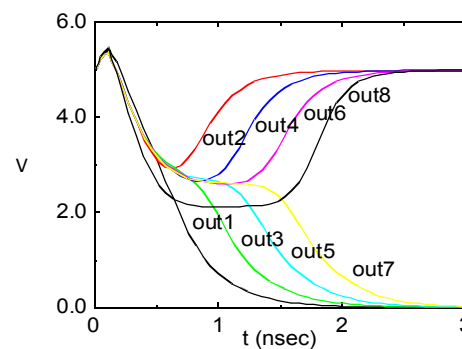
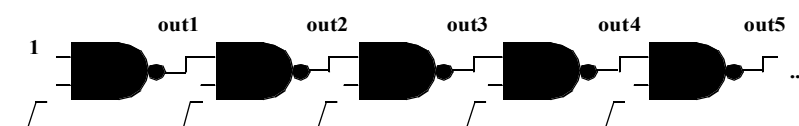
Glitches

- Glitch
 - Dynamic hazards
 - Useless behaviour
 - Important useless power



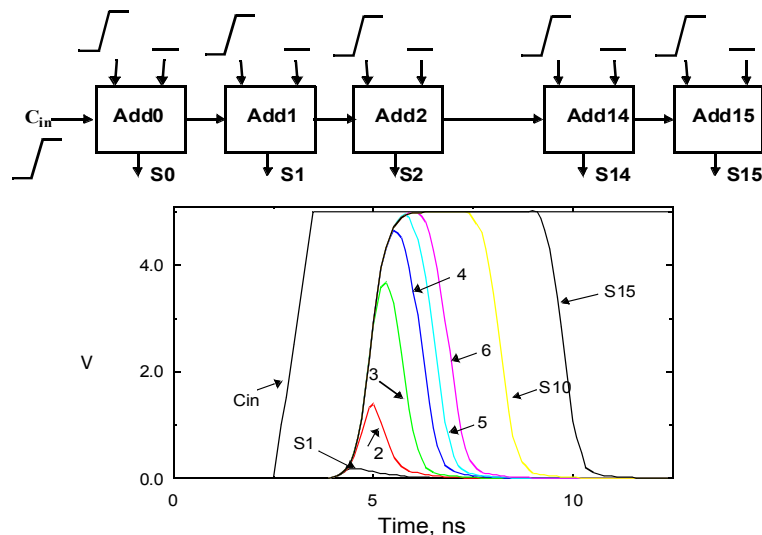
155

Example 1: Chain of NAND gates



156

Example 2: Adder



157

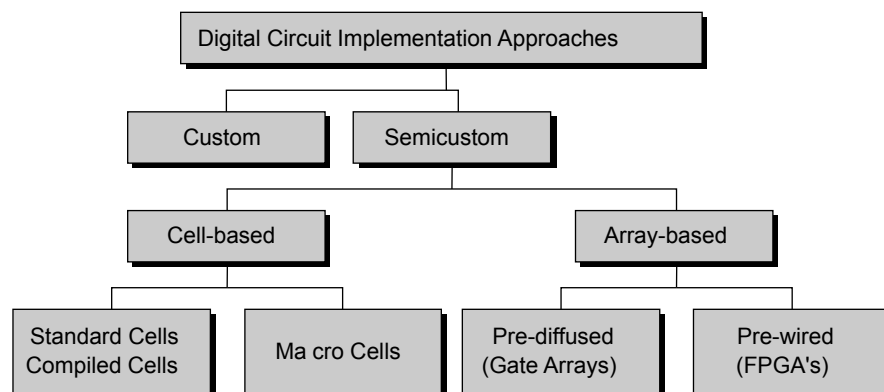
III. IC Design Methods

1. IC Classification
 1. Full-Custom
 2. Standard-Cells
 3. Gate Array
 4. Programmable Logic (CPLD, FPGA)
 5. Metrics
2. Design Methods and CAD Tools
 1. Design flow
 2. Top-Down or Bottom-Up
 3. CAD Tools
3. IC Specification
 1. Detailed specifications
 2. Outline of a specification document

158

III.1. IC Classification

- Trade-off between: flexibility, cost, design time (time-to-market), prototyping, foundry, tools



159

1. Full-Custom Circuits

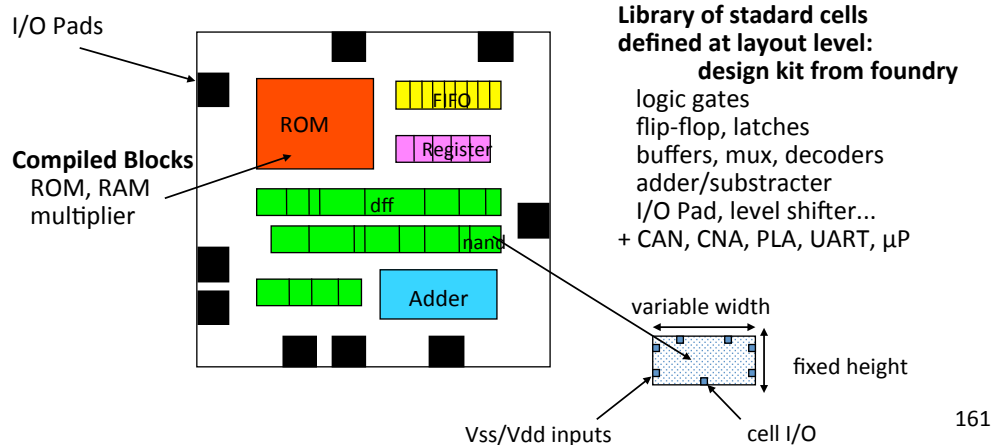
- Each transistor can be optimized
- Advantages
 - density, performance, power
- Drawbacks
 - cost
 - design time (6/17 trans./day)
 - re-engineering cost
- Used in:
 - processors
 - memory
 - spatial domain



160

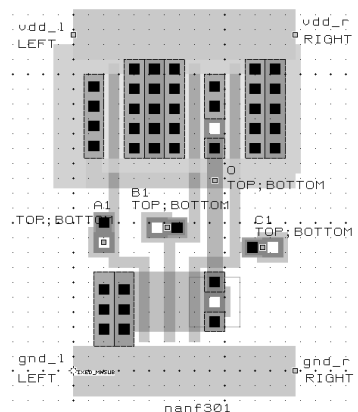
2. Cell-Based (or Standard Cells) Circuits

- Based on cells in a technological library
- Use place&route for metal routing



Standard Cell Library

- Cell example: 3-input NAND



AMS **NA3** CUB 0.6 μm CMOS

NA3 is a 3-input NAND gate with 1x drive strength.

Truth Table

A	B	C	Q
L	X	X	H
X	L	X	H
X	X	L	H
H	H	H	L



Capacitance

	C _i (pF)
A	0.036
B	0.036
C	0.041

Area

0.54 $mils^2$

Power

4.27 $\mu W/MHz$

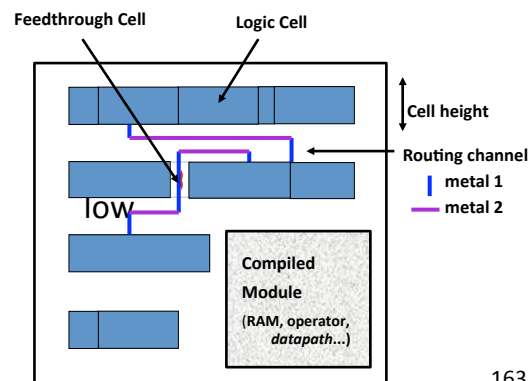
Delay [ns] = $t_{pd} = 6(SL \cdot L)$ with SL = Input Slope [ns], L = Output Load [pF]
Output Slope [ns] = $op_sl = 1/L$ with L = Output Load [pF]

AC Characteristics: $T_j = 25^\circ C$, VDD = 5V, Typical Process

Characteristics		SL = 0.1			SL = 2.0		
		L = 0.1	L = 0.7	L = 1.0	L = 0.1	L = 0.7	L = 1.0
Delay A to Q	t_{pdA}	0.30	1.28	1.81	0.65	1.49	1.93
	t_{pdA}	0.24	1.03	1.48	0.13	0.91	1.27
Delay B to Q	t_{pdB}	0.28	1.25	1.78	0.60	1.45	1.90
	t_{pdB}	0.22	1.03	1.40	0.22	1.01	1.38
Delay C to Q	t_{pdC}	0.25	1.23	1.76	0.53	1.41	1.87
	t_{pdC}	0.20	1.02	1.40	0.29	1.12	1.50
Output Slope A to Q	op_slA	0.85	3.93	5.43	1.27	4.01	5.51
	op_slA	0.62	3.18	4.50	1.01	3.31	4.62
Output Slope B to Q	op_slB	0.77	3.81	5.37	1.20	3.93	5.41
	op_slB	0.62	3.16	4.52	1.06	3.32	4.63
Output Slope C to Q	op_slC	0.72	3.97	5.50	1.16	4.03	5.49
	op_slC	0.63	3.18	4.47	1.10	3.42	4.53

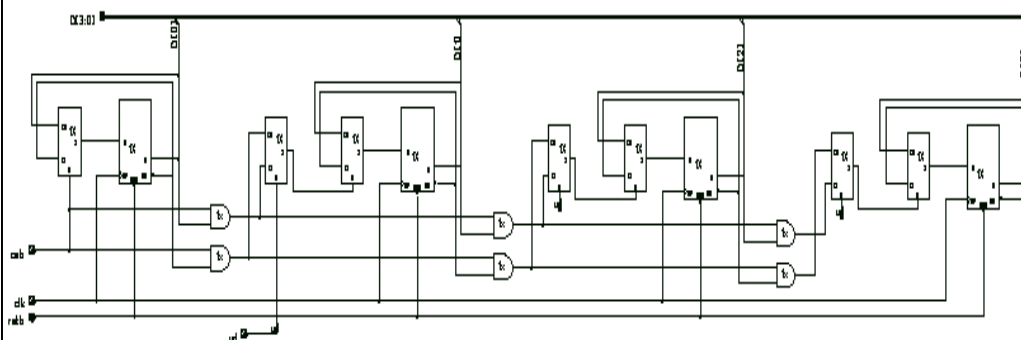
Cell-Based (or Standard Cells) Circuits

- Advantages
 - Reduction of cost and design time
 - reuse pre-designed cells
 - design kit
 - Chip includes only needed cells
 - But challenge is on place and route
- Drawback
 - Cost is still high for low volume
- Usually combines Full-Custom and Standard-Cells



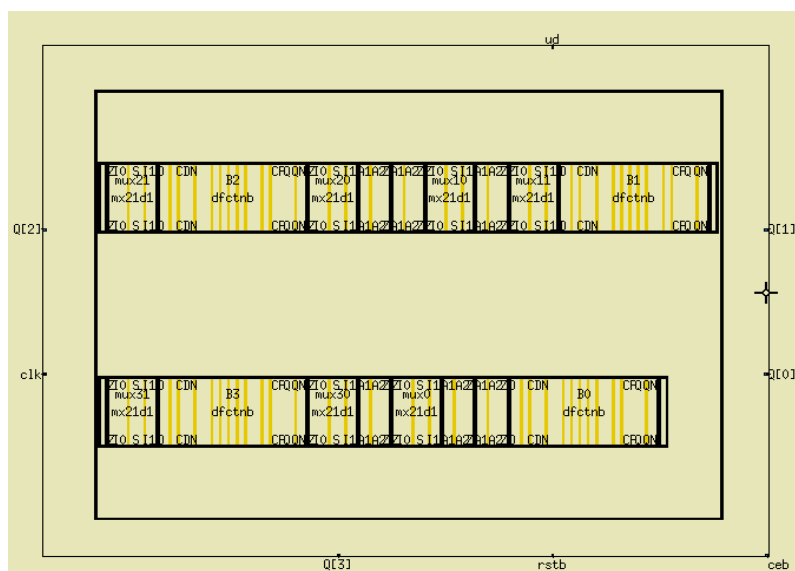
Place and Route

- 4-Bit Counter (up/down) with enable



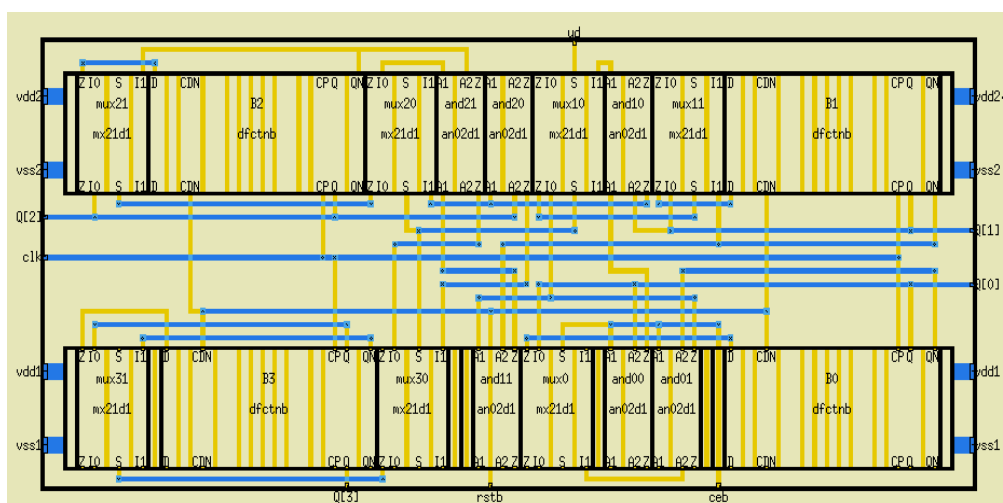
164

Placement



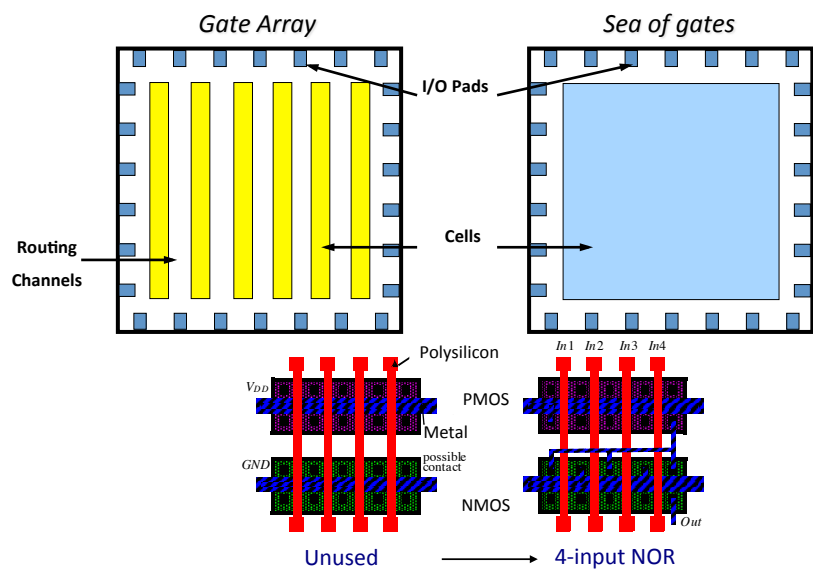
165

Routing



166

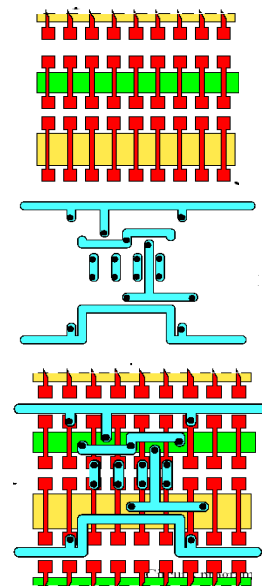
3. Gate Array



167

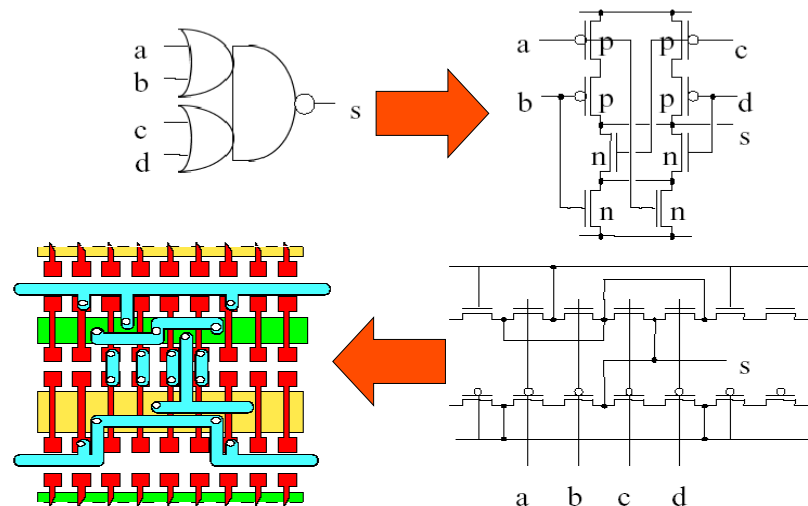
Gate Array

- High-volume production of transistors
 - Regular and dense organization
 - Fabrication on wafers
 - Reduces cost of fabrication
- Customized metal levels to match design of customer
- Fabrication of the customized chip at lower volume and cost



168

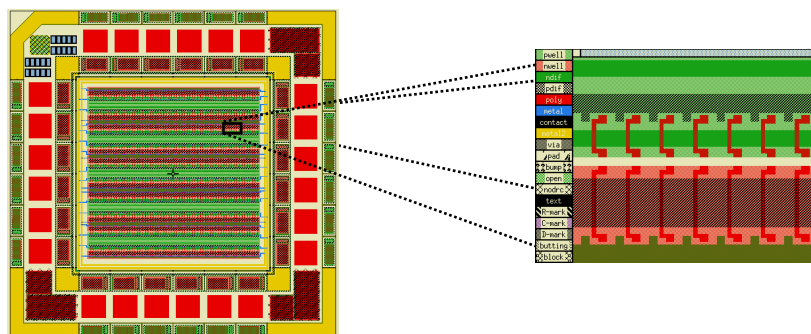
Gate Array



169

Gate Array

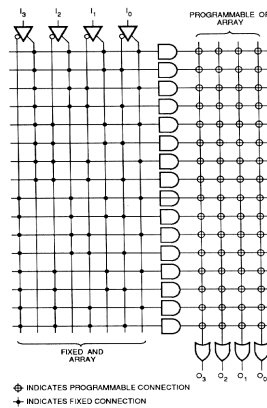
- Advantage: cost and design time
- Drawbacks: lower integration density
 - Unused transistors
 - Number of transistors is fixed



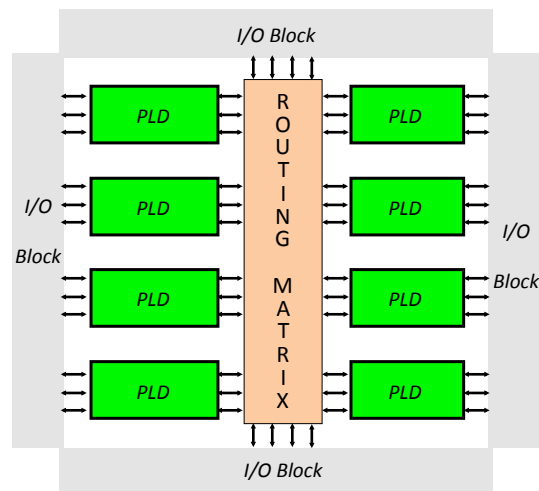
170

4. PLD: Programmable Logic Devices

PLD: Programmable Logic Device

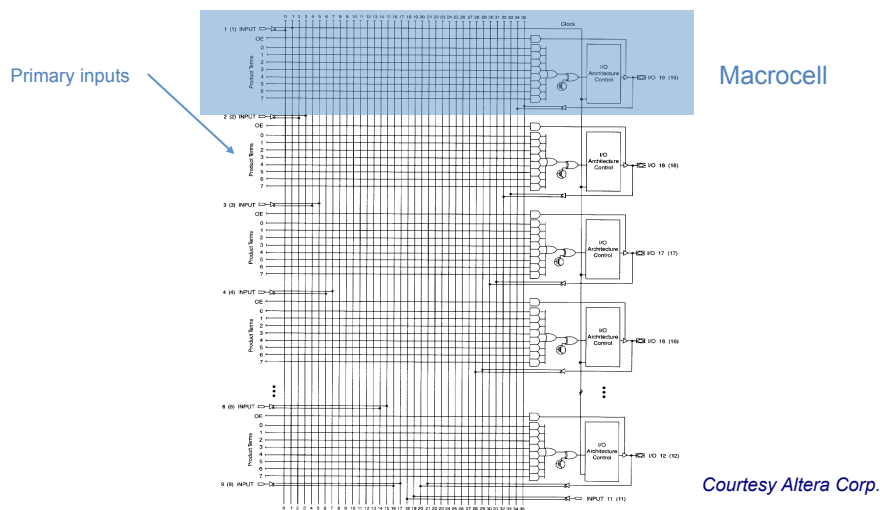


CPLD: Complex PLD



171

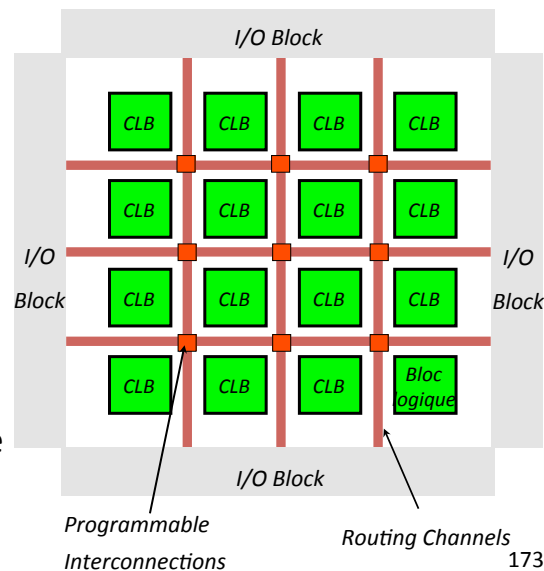
EPLD (Altera)



172

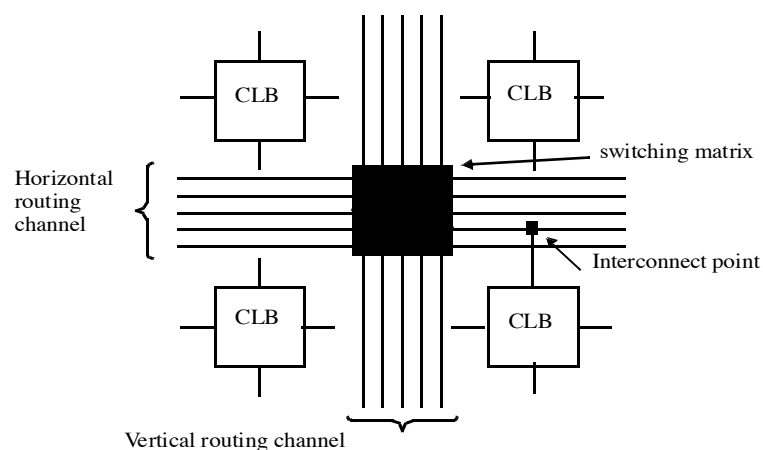
FPGA: Field-Programmable Gate Arrays

- Matrix of Logic Blocks and Programmable Interconnects
- FPGA types
 - SRAM FPGA
 - Antifuse/Flash FPGA
- Examples:
 - Altera: Stratix, Cyclone
 - Xilinx: Virtex, Spartan
 - Actel/Microsemi: Igloo



SRAM-based FPGAs

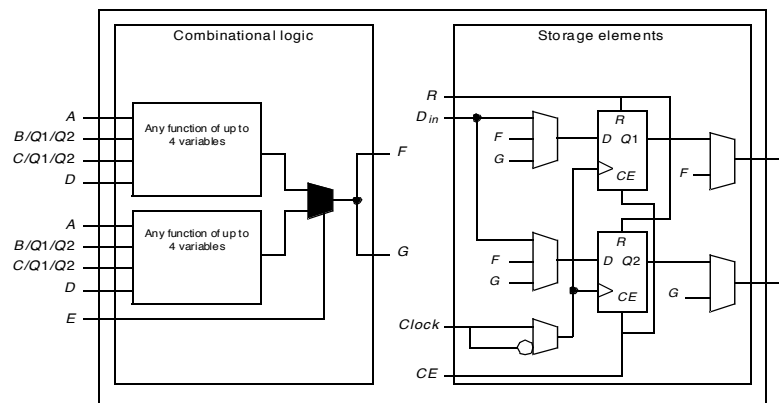
- Xilinx Architecture



174

SRAM-based FPGAs

- Xilinx Basic Cell (CLB)

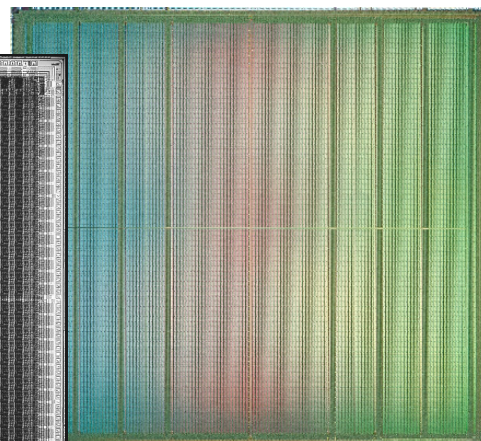
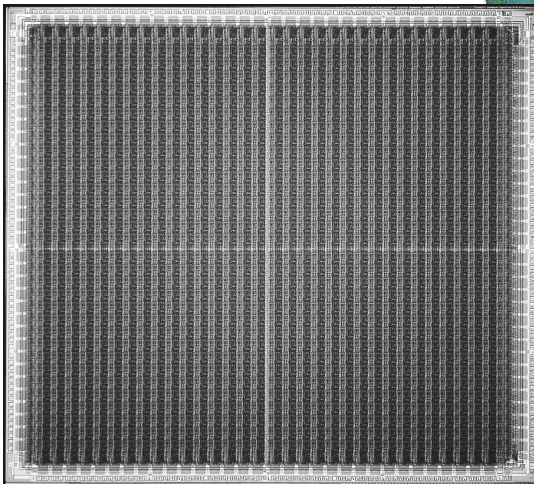


Courtesy of Xilinx

175

SRAM-based FPGAs

Xilinx XC4025



Xilinx Virtex

176

FPGA Xilinx VIRTEX 7

		Virtex-7 FPGAs Optimized for Highest System Performance and Capacity (1.0V, 0.9V)						
		Part Number	XC7V285T	XC7V450T	XC7V585T	XC7V855T	XC7V1500T	XC7V2000T
		EasyPath™ Cost Reduction Solutions ⁽¹⁾	XC7V285T	XC7V450T	XC7V585T	XC7V855T	XC7V1500T	XC7V2000T
Logic Resources	Slices ⁽²⁾	44,700	70,450	91,050	133,350	229,050	305,400	
	Logic Cells ⁽³⁾	286,080	450,880	582,720	853,440	1,465,920	1,954,560	
	CLB Flip-Flops	357,600	563,600	728,400	1,066,800	1,832,400	2,443,200	
Memory Resources	Maximum Distributed RAM (Kbits)	3,475	5,388	6,938	10,313	16,163	21,550	
	Block RAM/FIFO w/ ECC (36Kbits each)	410	615	795	1,155	969	1,292	
	Total Block RAM (Kbits)	14,760	22,140	28,620	41,580	34,884	46,512	
Clock Resources	Mixed Mode Clock Managers (MMCM)	14	14	18	18	18	24	
I/O Resources	Maximum Single-Ended I/O	700	700	850	850	850	1,200	
	Maximum Differential I/O Pairs	336	336	408	408	408	576	
	DSP48E1 Slices	700	980	1,260	1,800	1,620	2,160	
Embedded Hard IP Resources	PCI Express Interface Blocks ⁽⁴⁾	2	3	3	3	3	4	
	Analog Front End (XADC) / System Monitor Blocks	1	1	1	1	1	1	
	Configuration AES / HMAC Blocks	1	1	1	1	1	1	
	GTX 10.3125Gbps Transceivers	28	28	36	36	36	36	
	GTZ 13.1Gbps Transceivers	—	—	—	—	—	—	
	GTZ 28.05Gbps Transceivers	—	—	—	—	—	—	
Speed Grades	Commercial	-1L, -1, -2, -3	-1L, -1, -2, -3	-1L, -1, -2, -3	-1L, -1, -2, -3	-1L, -1, -2	-1L, -1, -2	
	Industrial	-1L, -1, -2	-1L, -1, -2	-1L, -1, -2	-1L, -1, -2	-1L, -1	-1L, -1	
Configuration	Configuration Memory (Mbits)	75.4	115.4	148.4	214.9	314.4	419.1	
		Package ⁽⁷⁾	Area		Available User I/O: 3.3V capable SelectIO™ Pins ⁽⁵⁾ , 1.8V SelectIO Pins ⁽⁶⁾ (GTX Transceivers)			

>2M CLBs

>45MB RAM

>4000 Mult.

177

177

5. Metrics

	Full Custom	Standard Cell	Gate Array	FPGA
Integration Density	✓ ✓	✓	~	×
Performances	✓ ✓	✓	~	×
Generic/Reuse	×	✓ ✓	×	✓ ✓
Design Time	×	×	~	✓ ✓
CAD Tools Complexity	×	×	×	~
Cost (low volume)	×	×	~	✓ ✓
Cost (high volume)	✓ ✓	✓	×	×

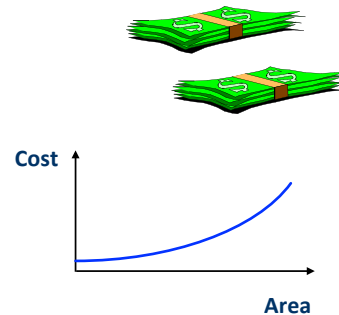
178

Cost

- Total Cost = NRE + UP x volume
 - NRE: Non Recurring Expenses
 - UP: Unit Price
- Cost per chip = NRE/volume + UP
- Yield: $\rho = (1 + N_0 \cdot A)^{-n}$
 - A: Chip Area – A_{tot} : Wafer Area
 - n: number of masks
 - N_0 : defect density per mask
- Unit Price = Wafer Price / (number of chip x ρ)

$$= \text{Wafer Price} (1 + N_0 \cdot A)^n (A/A_{tot})$$

$$= K \cdot A \cdot (1 + n \cdot N_0 \cdot A) \quad \text{if } N_0 \cdot A \ll 1$$



179

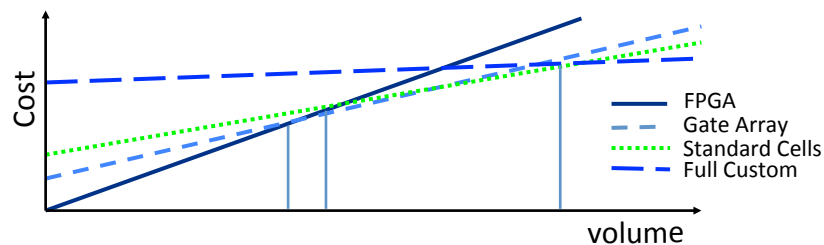
NRE: Non Recurring Expenses

- Mask set: up to several M\$
- Fabrication: circuit fabrication costs; eventually cost of P&R and DRC; test options, yield options, performance, analog precision, etc.
- CAD Tools: 10-100k€ plus maintenance
- Hardware: servers, disks, etc.
- Engineers...
- Re-design costs...

180

Cost

- Total Cost = NRE + UP x volume
 - NRE: Non Recurring Expenses
 - UP: Unit Price
- Cost per chip = NRE/volume + UP



181

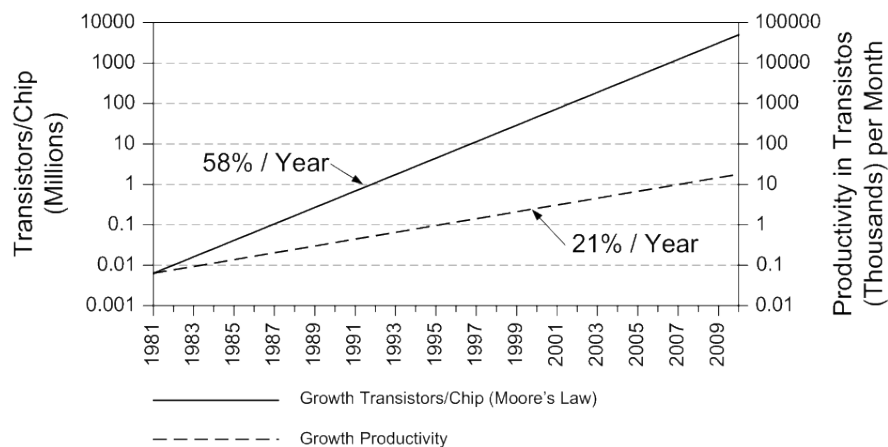
III. IC Design Methods

1. IC Classification
 1. Full-Custom
 2. Standard-Cells
 3. Gate Array
 4. Programmable Logic (CPLD, FPGA)
 5. Metrics
2. Design Methods and CAD Tools
 1. Design flow
 2. Top-Down or Bottom-Up
 3. CAD Tools
3. IC Specification
 1. Detailed specifications
 2. Outline of a specification document

182

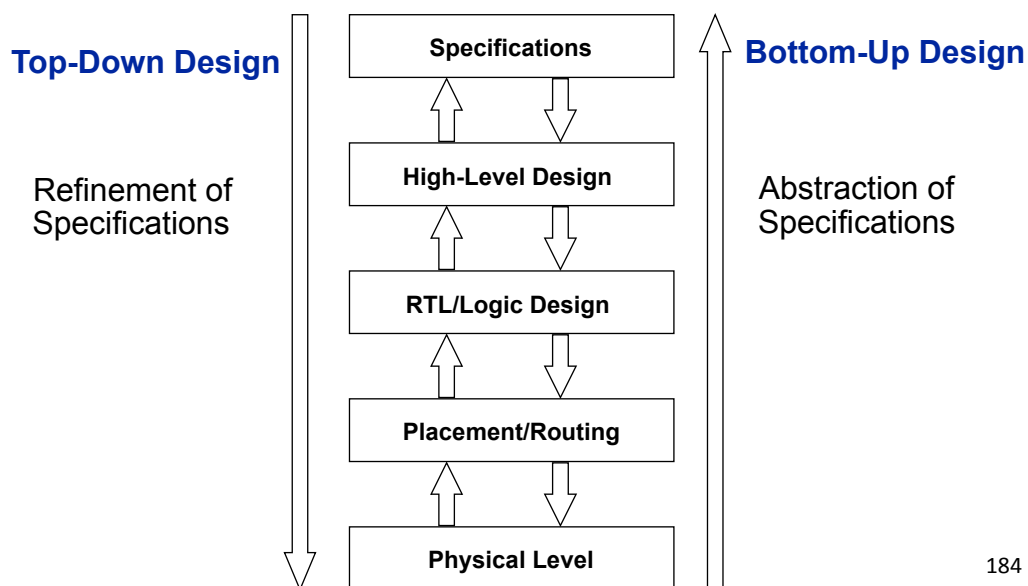
Design Productivity Gap

- Productivity is limited by design methods and CAD tools



183

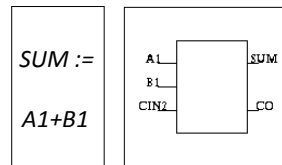
Design Flow



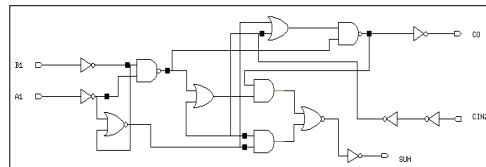
184

Design Flow: from algorithm to circuit

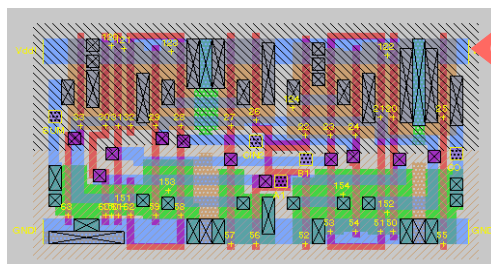
Register Transfer Level (RTL)



Gate Level

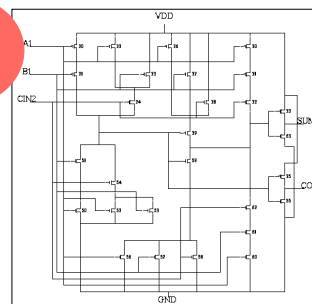


Algorithm



Layout Level

Circuit



Transistor or Circuit Level

185

Hardware vs. Software

Hardware

- Layout-Level
- Logic-Level
- Register-Transfer Level (VHDL, Verilog)
- Algorithmic Level (C/C++, System Verilog)
- System Level (e.g. UML, Matlab/Simulink)

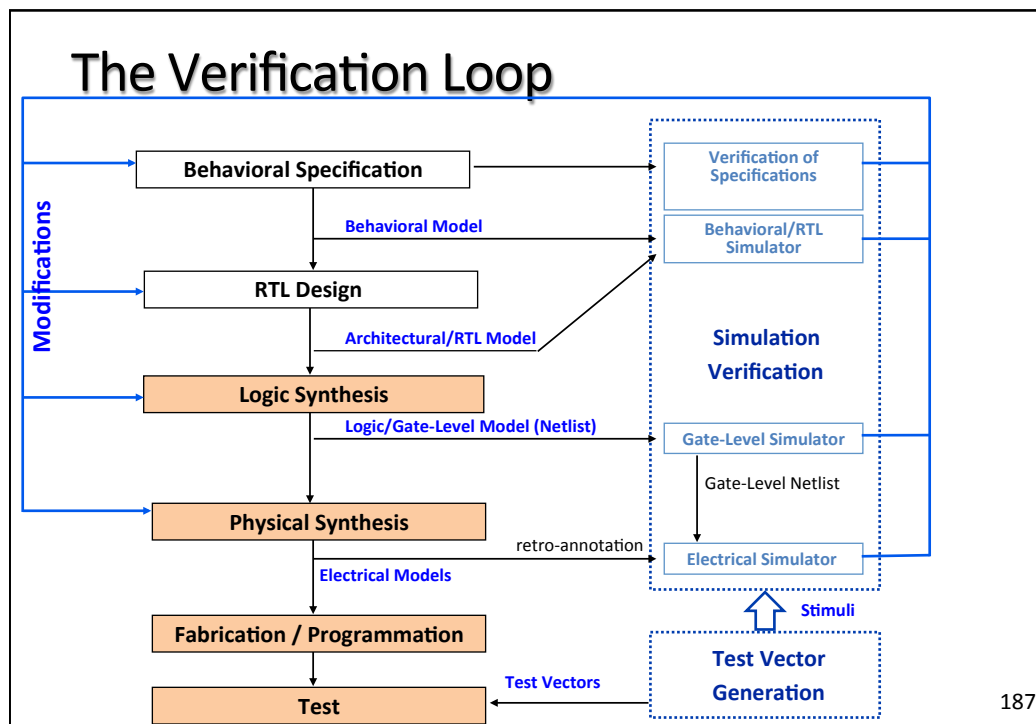
Abstraction Level



Software

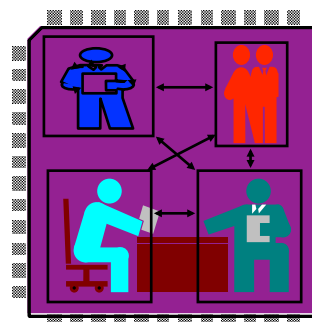
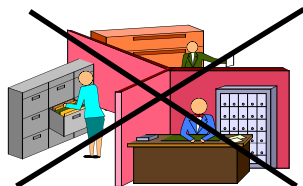
- Binary Code
- Assembly Code
- Machine Dependent Languages (e.g. C)
- Virtual Machine (e.g. Java)
- System Specifications (e.g. UML)

186



IC Design: a Real Team Work

- Teams of several engineers to design an IC (10-100)
 - Accurate task/block decomposition
 - Clear definition of interface and data exchange
 - Task/work distribution
 - Blocks to People
 - Front End, Back End, Verification Tasks
 - Frequent Meetings for
 - Specification Updates
 - Design Follow-Up



One Person One Block
Each Block Communicates with Others¹⁸⁸

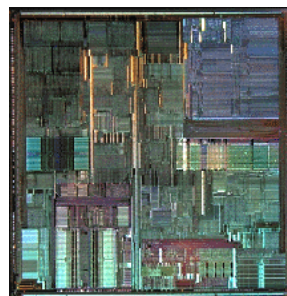
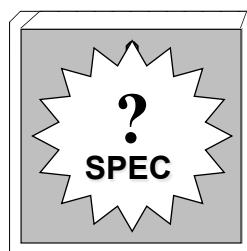
III. IC Design Methods

1. IC Classification
 1. Full-Custom
 2. Standard-Cells
 3. Gate Array
 4. Programmable Logic (CPLD, FPGA)
 5. Metrics
2. Design Methods and CAD Tools
 1. Design flow
 2. Top-Down or Bottom-Up
 3. CAD Tools
3. IC Specification
 1. Detailed specifications
 2. Outline of a specification document

189

Spécification d'un ASIC

- Le terme **spécification** regroupe toutes les informations qui caractérisent de l'extérieur le composant à réaliser. Les “spéc” sont indépendantes de l'utilisation qui est faite du circuit. Elles ont pour but de décrire ce que doit faire le composant (le QUOI) et pas du tout comment il le fait (le COMMENT).
- Généralement sous forme papier accompagné de modèles



190

Nature de la spécification d'un circuit

- Spécifications fonctionnelles
 - Description des fonctions que doit assurer le circuit (ou le bloc)
 - Equation logique, table de vérité, chronogramme
 - Diagramme état/transition, Grafcet, Statechart (extension du diagramme d'état au parallélisme)
 - Réseau de Petri (comportement temporel)
 - Modèle mathématique, signal, commande
 - Description algorithmique
- Spécifications opératoires
 - Manière dont une fonction doit opérer, conditions et domaines de fonctionnement
 - Renseignements sur les grandeurs ou données utilisées dans les spécifications fonctionnelles (type, domaine de définition, précision)
 - Informations pour guider les concepteurs dans le choix des solutions à mettre en œuvre (expériences précédentes dans l'entreprise ou ailleurs)
 - Test à opérer sur le circuit
- Spécifications technologiques
 - Renseignements en rapport avec la réalisation matérielle
 - Définition électrique des E/S
 - Performances, contraintes
 - Spécifications liées à la réalisation (taille, coût, technologie, type de boîtier,...)
 - Contraintes de l'environnement
 - Qualité de test

191

Spécification détaillée d'un ASIC

- Une spécification détaillée (Detailed Design Specification) est un document écrit qui regroupe
 - La spécification de la définition
 - La notice descriptive de fonctionnement
- Elle doit fournir toutes les informations utiles aux concepteurs des cartes utilisatrices, ainsi qu'aux concepteurs de logiciels.

192

Plan type de spécification détaillée

- 1 Introduction
 - Rappel de l'utilisation du circuit sur la carte
 - Rôle principal et fonctions
- 2 Environnement du circuit
 - Présentation générale de l'environnement du circuit sur la carte
- 3 Organisation générale du circuit
 - Interfaces externes
 - Présentation générale des fonctions
 - Synoptique générale du circuit faisant apparaître les blocs fonctionnels
 - Description des liaisons inter-blocs
- 4 Fonctions réalisées
 - Description détaillée des fonctions réalisées par le circuit
 - fonctions d'entrées-sorties
 - fonction d'initialisation
 - fonctions pour le test du composant
 - fonctions pour le test en sortie de fabrication des cartes utilisant le circuit

193

Plan type (suite)

- 5 Caractéristiques matérielles
 - Interface physique
 - Packaging
 - Brochage, description des signaux d'E/S et des alimentations
 - Technologie des E/S
 - Chronogramme et Timings
 - Chronogrammes théoriques associés aux diverses fonctions
 - Timings à respecter
 - Caractéristiques électriques
 - Limites électriques, conditions transitoires et opérationnelles
 - Caractéristiques statiques et dynamiques des E/S (alimentations, tension, courant, capacités de charge, Slew Rate Control,...)
 - Découplage d'alimentation à prévoir sur la carte
- 6 Interface Logiciel
 - Synthèse des informations relatives à l'interface matériel/logiciel
 - Description des registres et compteurs accessibles et de leur mode d'accès
 - L'utilisateur logiciel doit pouvoir se limiter à ce paragraphe pour le développement des logiciels

194

Spécification de réalisation

- Dans la phase de conception des blocs, cette spécification précise aux concepteurs les directives de réalisation
- Décrit l'architecture interne et fournit une description détaillée de chacun des blocs constituant le circuit
- Ce document est indispensable pour les circuit dont la complexité nécessite plusieurs concepteurs
- En cours de conception, la spécification fait l'objet de mises à jour

Plan type

- Présentation générale de la découpe en blocs
 - Diagramme général du circuit découpé
 - Interfaces inter-blocs ou avec l'extérieur
 - Bus et liaisons de contrôle principaux, distribution d'horloge interne, cycles d'échanges entre blocs
 - Pour chaque bloc :
 - description succincte de la fonction réalisée
 - estimation de la complexité
 - mémoires ou cellules spécifiques nécessaires (capacité, temps de cycle, simple/multi port...)
 - fréquence moyenne de fonctionnement et taux d'activité

195

Plan type (suite)

- Contraintes de réalisation
 - Boîtier : type de montage sur carte (soudé, support)
 - Nombre et type d'E/S
 - Fréquence aux accès
 - Contraintes principales de timing
 - Bilan des mémoires et/ou cellules nécessaires
 - Technologie (CMOS, bipolaire, ECL,...)
 - Référence fondeur du circuit
- Description détaillée des interfaces inter-blocs
 - Fonctions et chronogrammes de chaque bus ou liaison de contrôle interne
- Description détaillée de chaque bloc
 - Fonctions réalisées
 - Accès E/S
 - Structure interne du bloc
 - Complexité du bloc (nombre de portes)

196

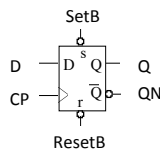
IV Synchronous Design of IC

1. Synchronous Design Rules and Principles
 1. General Issues and Parameters: critical path, clock skew
 2. Synchronous Rules (Ten Commandments)
 3. Recommended and Non-Recommended Circuits
2. Finite State Machine (FSM) plus Datapath Model
 1. Synchronous Models
 2. State Diagrams
 3. Moore/Mealy Machines
3. Arithmetic Operators
 1. Adders and Substracteurs
 2. Multipliers

197

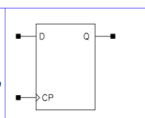
Timing Parameters

- D Flip-Flop
 - Setup Time: T_{setup}
 - Hold Time: T_{hold}
 - Propagation Time: T_p
 - on Clock and Reset



FD1QLL
FD1QLLP
FD1QLLX4

Function: Function = D Flip-Flop with 1 Phase Positive Edge Triggered Clock, Q Output Only



Truth Table

IQ	Q
IQ	IQ

Truth Table

D	CP	IQ	IQ
D	/	-	D
-	-	IQ	IQ

Physical Dimensions

Property	FD1QLL	FD1QLLP	FD1QLLX4
Area(μm ²)	28.241	28.241	30.258

Capacitance
picoFarads

Cell	Property	Best 1.32V -40C	Worst 1.08V 125C	Nominal 1.2V 25C
FD1QLL	CP Input Cap.	0.0032	0.0028	0.0030
FD1QLL	Q Max Load	0.160	0.160	0.160
FD1QLL	D Input Cap.	0.0023	0.0020	0.0021
FD1QLLP	Q Max Load	0.320	0.320	0.320
FD1QLLP	D Input Cap.	0.0022	0.0019	0.0021
FD1QLLP	CP Input Cap.	0.0032	0.0027	0.0029
FD1QLLX4	CP Input Cap.	0.0032	0.0027	0.0029
FD1QLLX4	Q Max Load	0.640	0.640	0.640
FD1QLLX4	D Input Cap.	0.0022	0.0019	0.0020

Propagation Delay
nanoSeconds, as a function of C (load in pF) and Tr (input transition time in nS)

Cell	Path	Event	Best 1.32V -40C	Worst 1.08V 125C	Nominal 1.2V 25C
FD1QLL	CP-Q	CP_Q (fall)	0.082 + 0.119*Tr + 1.221°C	0.195 + 0.179*Tr + 2.77°C	0.125 + 0.148*Tr + 1.731°C
FD1QLL	CP-Q	CP_Q (rise)	0.075 + 0.118*Tr + 1.672°C	0.178 + 0.180*Tr + 3.473°C	0.113 + 0.148*Tr + 2.408°C
FD1QLLP	CP-Q	CP_Q (fall)	0.087 + 0.121*Tr + 0.844°C	0.205 + 0.182*Tr + 1.428°C	0.133 + 0.150*Tr + 0.903°C
FD1QLLP	CP-Q	CP_Q (rise)	0.079 + 0.120*Tr + 0.836°C	0.189 + 0.182*Tr + 1.727°C	0.120 + 0.150*Tr + 1.198°C
FD1QLLX4	CP-Q	CP_Q (fall)	0.111 + 0.122*Tr + 0.342°C	0.267 + 0.183*Tr + 0.760°C	0.173 + 0.152*Tr + 0.482°C
FD1QLLX4	CP-Q	CP_Q (rise)	0.093 + 0.121*Tr + 0.485°C	0.224 + 0.184*Tr + 0.891°C	0.141 + 0.151*Tr + 0.612°C

Power

- Leakage and Dynamic Power

Average Leakage Power
picoWatts

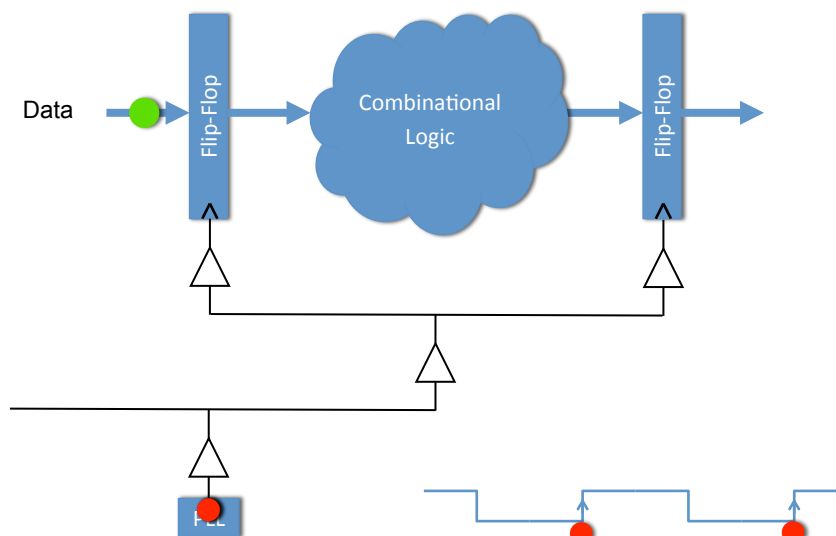
Cell	bc_1.32V_25C	bc_1.32V_125C
FD1QLL	10767.200	200860.000
FD1QLLP	12226.500	235294.000
FD1QLLX4	15436.600	308319.000

Internal Energy at minimum output load
uW/MHz as a function of Tr (input transition time in nS)

Cell	Cycle on pin	nom_1.20V_25C
FD1QLL	Q(related_pin:CP)	$0.029 + 0.013 \cdot Tr$
FD1QLL	D(max)	$0.020 + 0.007 \cdot Tr$
FD1QLL	CP(max)	$0.016 + 0.012 \cdot Tr$
FD1QLLP	Q(related_pin:CP)	$0.038 + 0.013 \cdot Tr$
FD1QLLP	D(max)	$0.021 + 0.006 \cdot Tr$
FD1QLLP	CP(max)	$0.017 + 0.012 \cdot Tr$
FD1QLLX4	Q(related_pin:CP)	$0.060 + 0.013 \cdot Tr$
FD1QLLX4	D(max)	$0.022 + 0.007 \cdot Tr$
FD1QLLX4	CP(max)	$0.017 + 0.012 \cdot Tr$

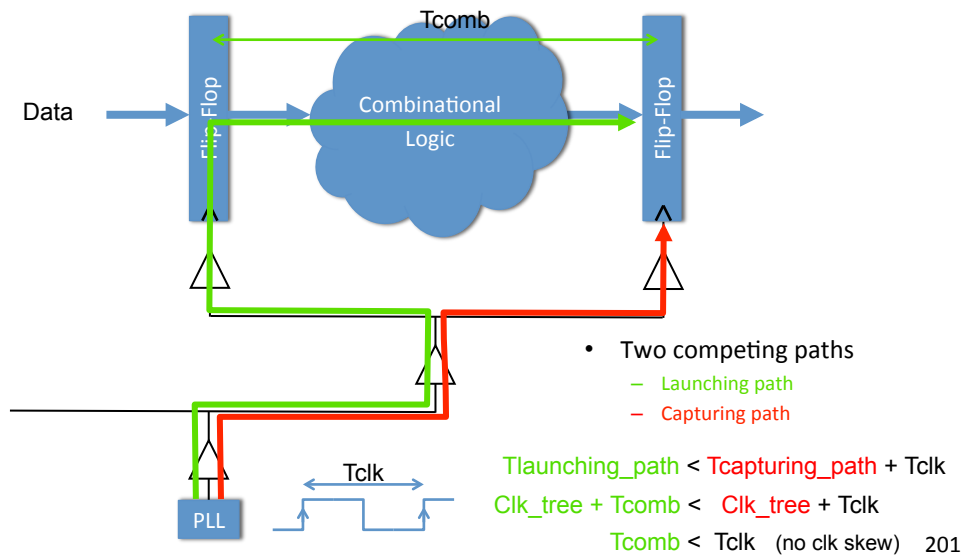
199

Synchronous Circuits



200

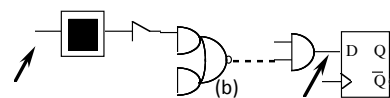
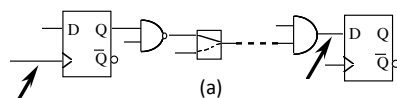
Synchronous Circuits



201

Critical Path

- All circuits have a **maximal frequency**, which is given by finding its **critical path**
 - Data must be stable when sampled by the clock
- Critical Path
 - Maximal combinational delay between:
 - Q output of a FF $\rightarrow$ logic $\rightarrow$ D input of a FF (a)
 - Q output of a FF $\rightarrow$ logic $\rightarrow$ output
 - input $\rightarrow$ logic $\rightarrow$ D input of a FF (b)
 - input $\rightarrow$ logic $\rightarrow$ output



202

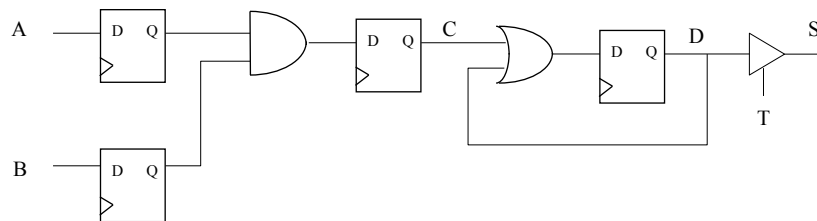
Critical Path

- T_{cp} : Critical Path Delay of the Logic

$$T_{cp} = \max_{\forall i} (D_i), \text{ with } D_i \text{ Delay of path } i$$

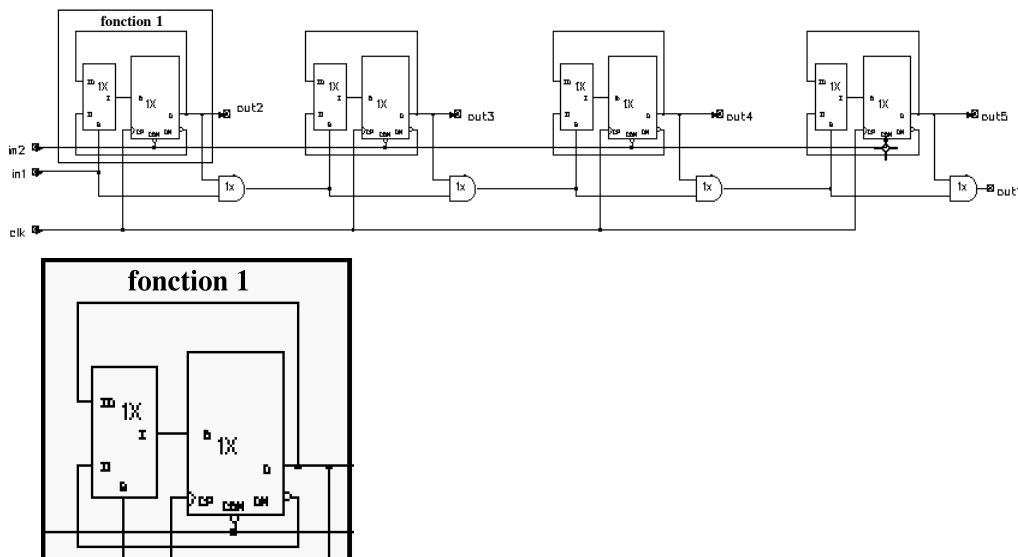
– For every logic path i in the considered circuit

- Maximal Frequency $F_{clk_max} = \frac{1}{T_{cp} + T_p + T_{setup}}$
- Example (1):



203

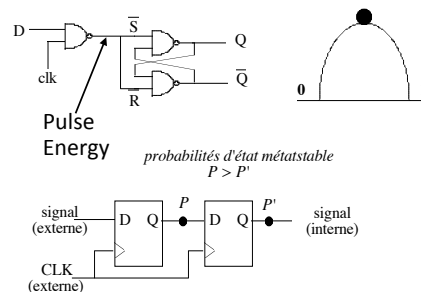
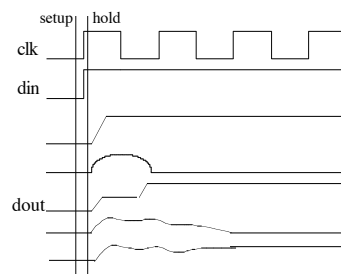
Example (2)



204

Meta-stability

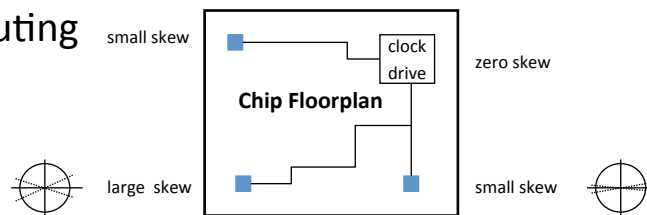
- Analog state between two logic states
 - Setup or Hold violation
 - Minimum signal pulse width violation
 - e.g. Asynchronous Reset
- Generates and propagates slow signal slopes, unstable states
- Cascade of flip-flop reduces meta-stability probability



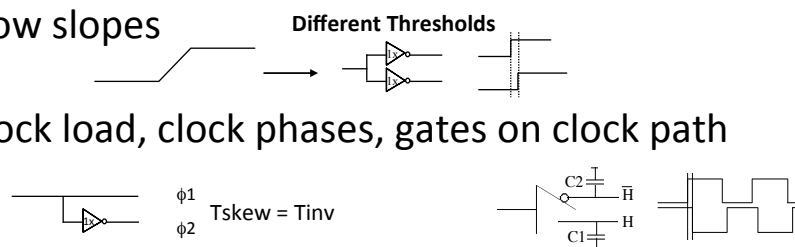
205

Clock Skew

- Every FF receives the clock edge at a different time
- Clock routing



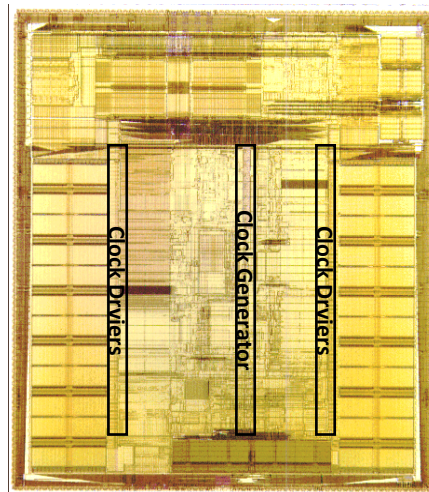
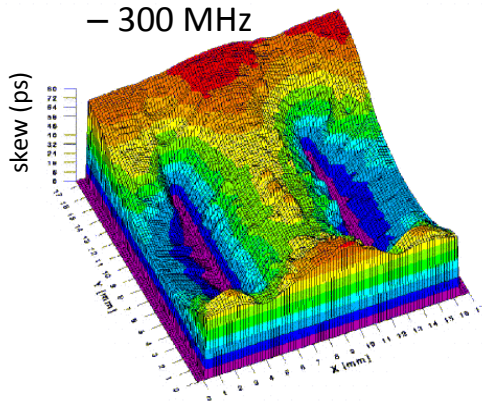
- Slow slopes
- Clock load, clock phases, gates on clock path



206

Clock Skew

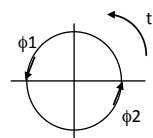
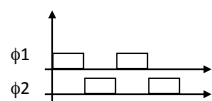
- DEC Alpha 21164 (1995)
 - 9.3 M. Transistors, 0.5 μ
 - 300 MHz



207

Clock Skew

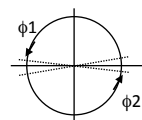
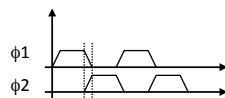
- Ideal Clock



$$\phi_1(t) \cdot \phi_2(t) = 0 \quad \forall t$$

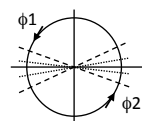
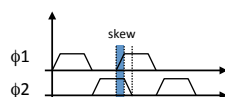
Timing Circle

- Slow Slopes



$$\begin{aligned} \phi_1(t) \cdot \phi_2(t) &= 0 && \text{on levels} \\ \phi_1(t) \cdot \phi_2(t) &\neq 0 && \text{on transitions} \end{aligned}$$

- With Clock Skew



Skew is the time during which $\phi_1(t) \cdot \phi_2(t) = 1$

208

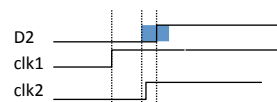
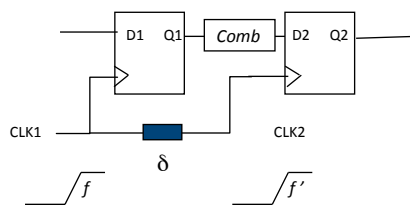
Clock Skew: problems

- Skew δ can be negative or positive

– Reduction of maximal frequency $f_{e_{\max}} = \frac{1}{T_{cc} + T_p + T_{setup} + \delta}$

– Maximal skew for circuit operation

- Worst case is when receiving edge arrives late
 - Edge f' of CLK2 should not violate hold time of D2
 - Race between data and clock



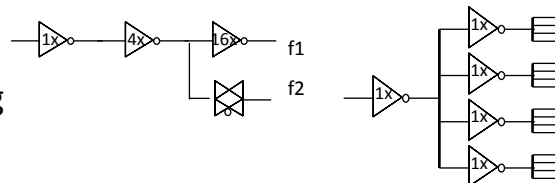
$$t_{p1} + \text{MIN}(t_{comb}) > \delta + t_{hold}$$

$$\delta < t_{p1} + \text{MIN}(t_{comb}) - t_{hold}$$

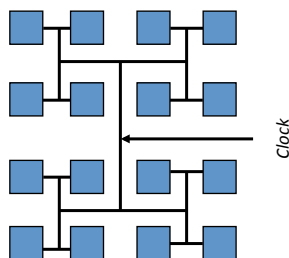
209

Clock Trees

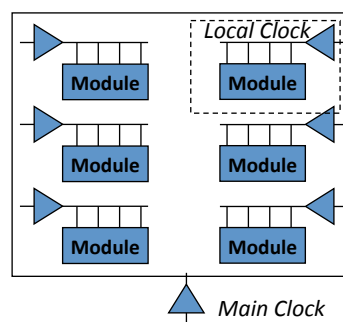
- Clock distribution
 - Geometric buffering
 - Tree-based



H-tree: constant skew in each block with equivalent number of flip-flops



Buffering: local reduction of skew



210

Synchronous Design Rules

- One **unique clock phase** is connected to all flip-flops of one clock domain
- One unique asynchronous reset is connected to all flip-flops

Don't touch to clock and asynchronous reset signals!

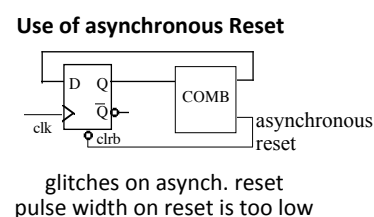
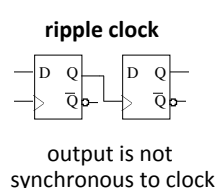
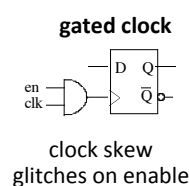


- A good clock tree and no slow slopes
- Some more rules
 - No combinatorial feedbacks
 - Centralized control of tri-state gates
 - Avoid using too many transmission gates in serial

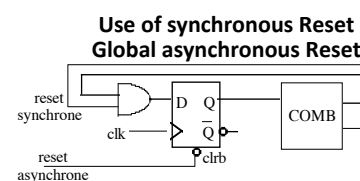
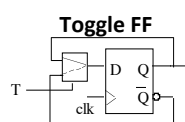
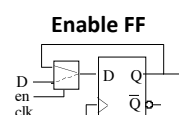
211

Synchronous Design Rules

- Non-recommended circuits



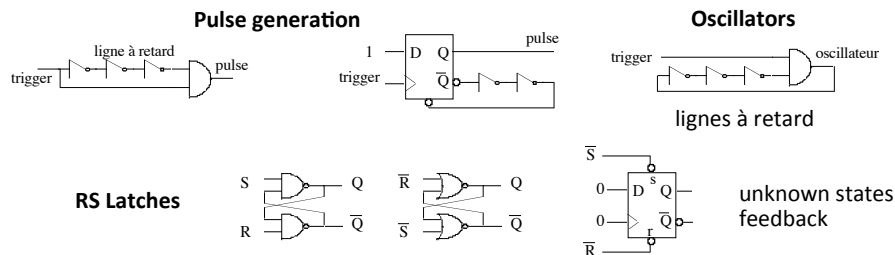
- Recommended circuits



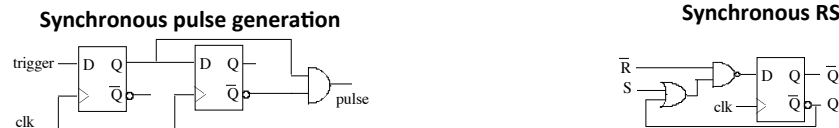
212

Synchronous Design Rules

• Non-recommended circuits



• Recommended circuits



213

Les dix commandements en conception de circuits intégrés numériques



1. Une seule horloge maîtresse tu auras et tu ne construiras point de fausses horloges à partir de circuits instables.
2. L'horloge tu ne manipuleras point et tu ne transmettras point à travers une porte logique, car cela causerait des aléas, des fausses transitions et des montées lentes.
3. Tu concevras tous les circuits selon les méthodes de conception synchrones à moins de pouvoir convaincre celui qui paie ton salaire, ou qui attribue ta note que, pour des raisons de rapidité, consommation de puissance ou publication d'articles scientifiques, les circuits synchrones ne peuvent faire l'affaire.
4. Tu ne t'associeras point avec des indésirables tels que les compteurs en cascade ("ripple counter") et les multivibrateurs un-coup ("oneshot") ou multivibrateur monostable, mais tu cultiveras des amitiés avec les compteurs Johnson, les compteurs pseudo-aléatoires et les bascules avec entrée d'activation ("enable").
5. Tous les éléments séquentiels tu raccorderas au **RESET** global afin que le circuit démarre toujours dans un état connu et défini et que tes simulations ne demeurent point indéfinies pour l'éternité.
6. Tu ne mélangeras point les mises à terre analogique et numérique, car une telle union ne peut mener qu'au désastre.
7. Il ne restera point impuni, celui qui laisse flotter des entrées CMOS.
8. Un reset asynchrone n'est pas prévu pour des tâches telles que remettre un compteur à zéro. Vraiment je te le dis, 6 circuits créés de cette manière se réinitialiseront correctement et sembleront t'apporter gloire et honneur, mais le septième échouera lamentablement et te précipitera dans la honte et la disgrâce.
9. Les entrées asynchrones impropres tu purgeras en les passant dans au moins une bascule D avant de leur donner accès aux pures variables d'états.
10. Quiconque comprend parfaitement les motivations des précédents commandements saura aussi quelles libertés peuvent être tolérées avec eux. Que celui qui les viole dans l'ignorance prenne garde!

214

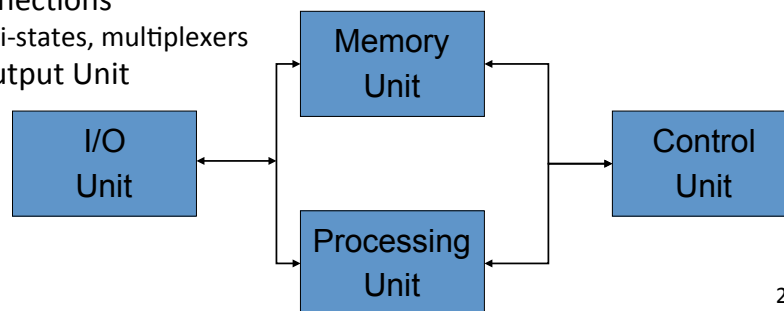
IV Synchronous Design of IC

1. Synchronous Design Rules and Principles
 1. General Issues and Parameters: critical path, clock skew
 2. Synchronous Rules (Ten Commandments)
 3. Recommended and Non-Recommended Circuits
2. Finite State Machine (FSM) plus Datapath Model
 1. Synchronous Models
 2. State Diagrams
 3. Moore/Mealy Machines
3. Arithmetic Operators
 1. Adders and Substracteurs
 2. Multipliers

215

General Architecture

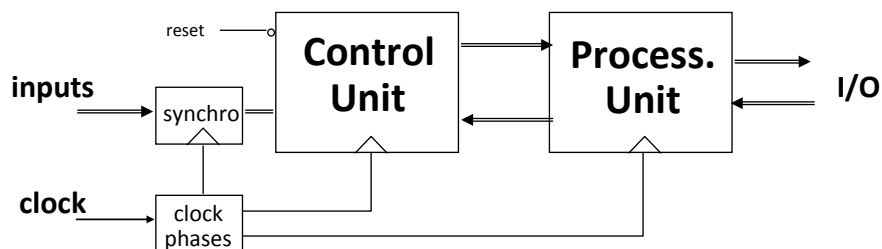
- Arithmetic Unit: datapath
 - adders, multipliers, shifters, comparators, etc.
- Memory Unit
 - RAM, ROM, register, register bank, FIFO, etc.
- Control Unit
 - Finite State Machines (FSM), counters, etc.
- Interconnections
 - Bus, tri-states, multiplexers
- Input/Output Unit



216

FSM + Datapath Model

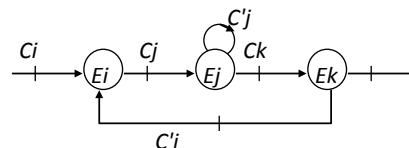
- Control Unit: Moore FSM (or synchronized Mealy)
- Synchronized Inputs for avoiding meta-stability
- Asynchronous Reset on the FSM is mandatory
- One unique or different clock phases



217

FSM Specification using State Diagram

- State Transition Diagram
- Nodes: FSM states
 - Number of nodes: state register size
- Transitions: Boolean conditions of states and inputs
 - All transitions at the output of a state are complementary
- State encoding: numerical value of state
 - binary, gray, Johnson
- FSM outputs depends on states (Moore) and on transitions (for Mealy)
- Clock is implicit



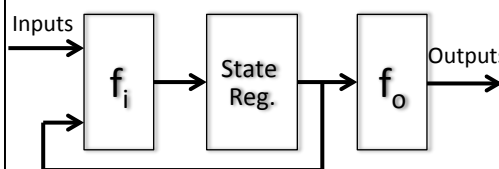
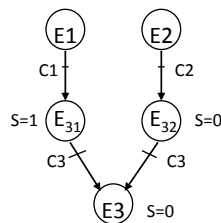
218

Moore/Mealy

- Moore

Outputs = $f_o(\text{Current State})$

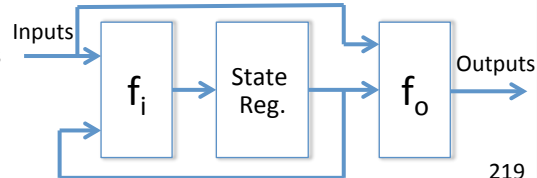
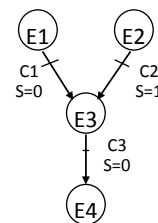
Next State = $f_1(\text{Current State, Inputs})$



- Mealy

Outputs = $f_o(\text{Current State, Inputs})$

Next State = $f_1(\text{Current State, Inputs})$



219

IV Synchronous Design of IC

1. Synchronous Design Rules and Principles

1. General Issues and Parameters: critical path, clock skew
2. Synchronous Rules (Ten Commandments)
3. Recommended and Non-Recommended Circuits

2. Finite State Machine (FSM) plus Datapath Model

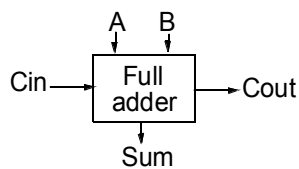
1. Synchronous Models
2. State Diagrams
3. Moore/Mealy Machines

3. Arithmetic Operators

1. Adders and Substracteurs
2. Multipliers

220

Full-Adder



$$S = A \oplus B \oplus C_i$$

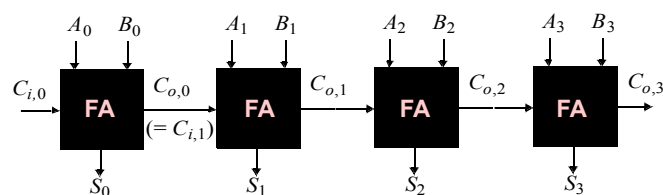
$$= A\bar{B}\bar{C}_i + \bar{A}B\bar{C}_i + \bar{A}\bar{B}C_i + ABC_i$$

$$C_o = AB + BC_i + AC_i$$

A	B	C_i	S	C_o	Carry status
0	0	0	0	0	delete
0	0	1	1	0	delete
0	1	0	1	0	propagate
0	1	1	0	1	propagate
1	0	0	1	0	propagate
1	0	1	0	1	propagate
1	1	0	0	1	generate
1	1	1	1	1	generate

221

Ripple-Carry Adder



Worst case delay linear with the number of bits

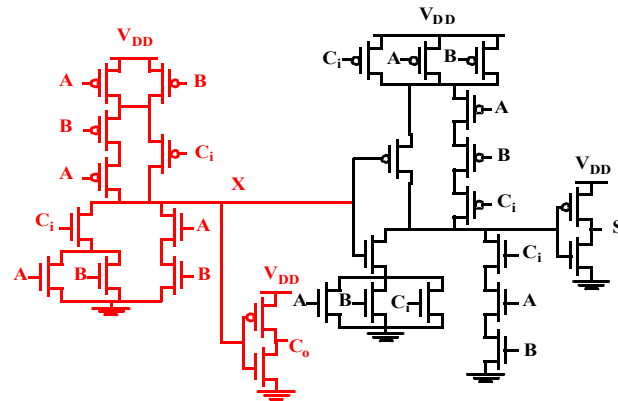
$$t_d = O(N)$$

$$t_{\text{adder}} \approx (N-1)t_{\text{carry}} + t_{\text{sum}}$$

Goal: Make the fastest possible carry path circuit

222

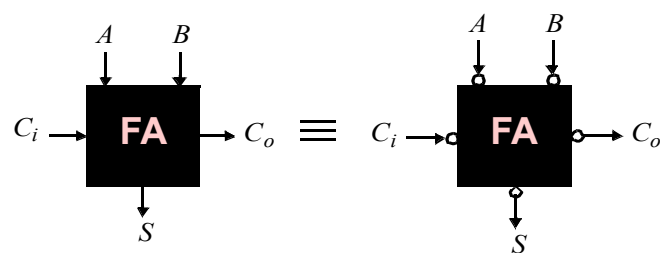
Complementary Static CMOS Full Adder



28 Transistors

223

Inversion Property



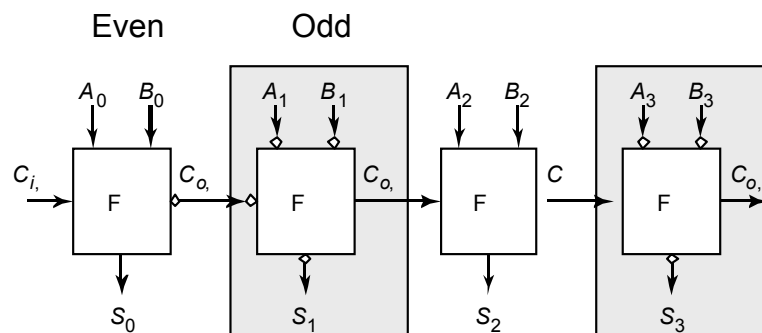
$$\bar{S}(A, B, C_i) = S(\bar{A}, \bar{B}, \bar{C}_i)$$

$$\bar{C}_o(A, B, C_i) = C_o(\bar{A}, \bar{B}, \bar{C}_i)$$

224

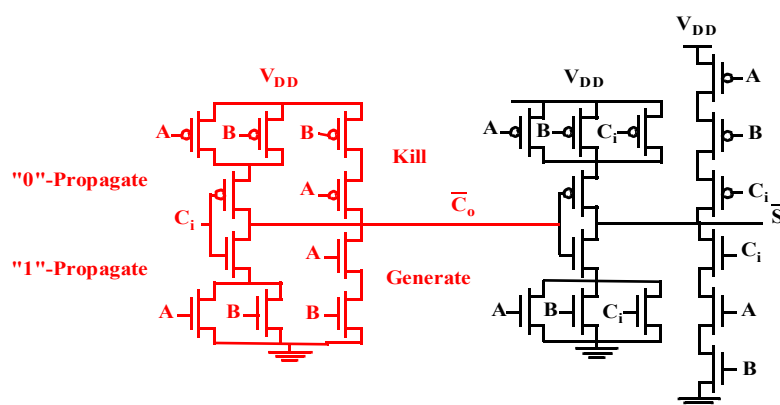
Ripple Carry Adder (V2)

- Exploiting inversion property
- Reduces propagation time



225

Mirror adder structure



24 transistors

226

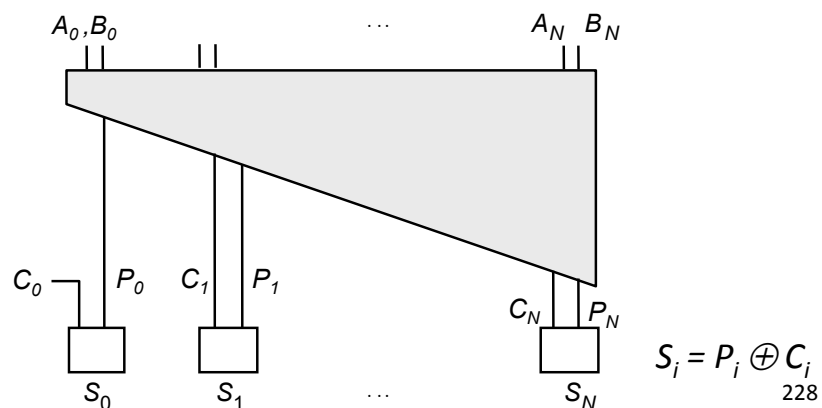
Adder Optimization

- Generation
 - if $A_i = B_i = 0$ then $C_{i+1} = 0$
 - if $A_i = B_i = 1$ then $C_{i+1} = 1$
 - when $A_i = B_i$ then C_{i+1} can be **generated**
- Propagation
 - when $A_i \neq B_i$ then $C_{i+1} = C_i$, carry is **propagated**
- Accelerating addition
 - $P_i = A_i \oplus B_i$ propagation at i
 - $G_i = A_i \cdot B_i$ generation at i
 - $S_i = A_i \oplus B_i \oplus C_i = P_i \oplus C_i$

227

Carry Look Ahead (CLA) Adder

- CLA: avoid long carry propagation through N FAs by calculating P_i, G_i, C_i from inputs for each FA i using logic cells



Carry Look Ahead (CLA) Adder

- At each stage i

- Compute $P_i = A_i \oplus B_i$

- Compute $G_i = A_i B_i$

- Compute C_i

$$C_i = G_{i-1} + P_{i-1} C_{i-1}$$

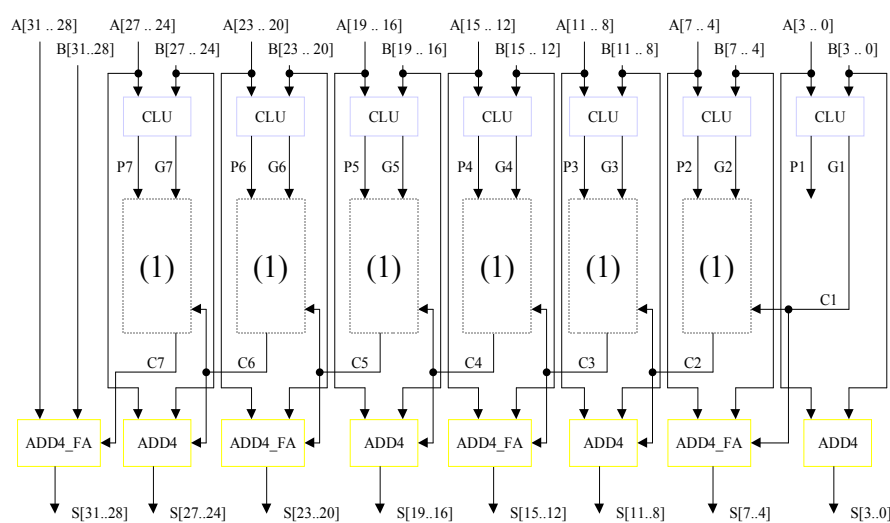
$$C_i = G_{i-1} + P_{i-1} (G_{i-2} + P_{i-2} C_{i-2}) = \dots$$

$$C_i = G_{i-1} + P_{i-1} G_{i-2} + P_{i-1} P_{i-2} G_{i-3} + \dots + P_{i-1} \dots P_1 G_0 + P_{i-1} \dots P_1 P_0 C_0$$

- Compute $S_i = P_i \oplus C_i$

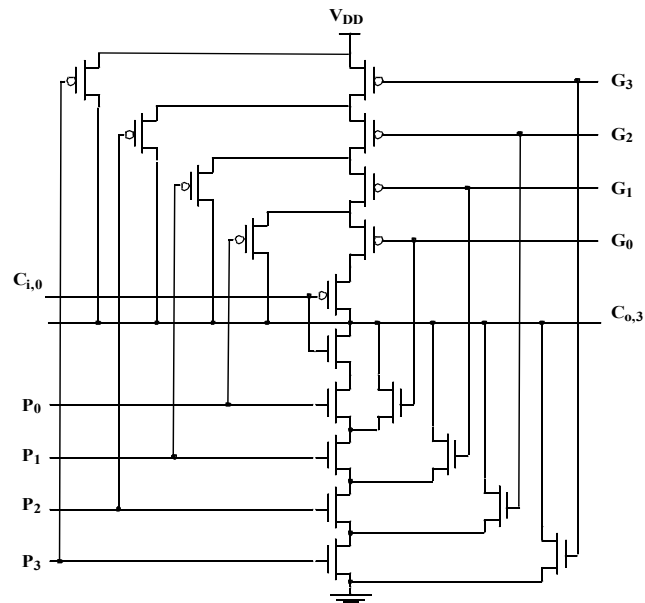
229

Carry Look Ahead (CLA) 32-bit Adder



230

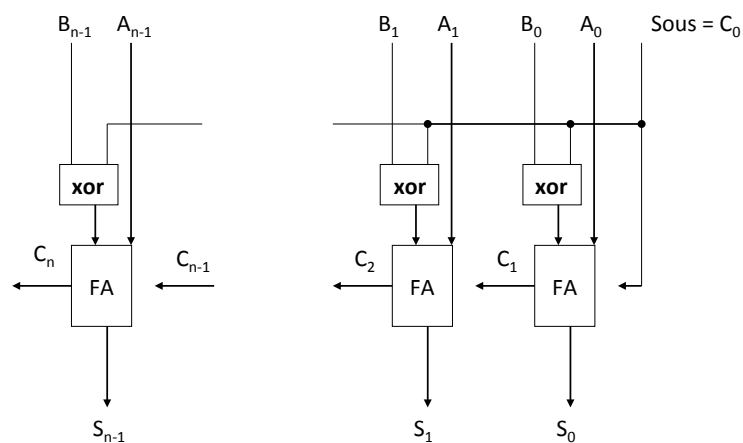
Look-Ahead Cell



231

Subtractor (two's complement)

- $A - B = A + \text{NOT}(B) + 1$



232

Multiplier

- Multiplication of two unsigned numbers

- Signed numbers

– A=-3, B=10

– a) P=290

– b) P=-30

$$\begin{array}{r}
 \times 11101 \\
 01010 \\
 \hline
 00000 \\
 11101 \\
 00000 \\
 11101 \\
 00000 \\
 \hline
 0100100010 \\
 a
 \end{array}$$

$$\begin{array}{r}
 \times 11101 \\
 01010 \\
 \hline
 0000000000 \\
 111111101 \\
 00000000 \\
 1111101 \\
 000000 \\
 \hline
 1111100010 \\
 b
 \end{array}$$

$$\begin{array}{r}
 \begin{array}{r}
 101010 \\
 \times 1011 \\
 \hline
 101010 \\
 101010 \\
 000000 \\
 + 101010 \\
 \hline
 111001110
 \end{array}
 \end{array}$$

Multiplicand
 Multiplier
 Partial products
 Result

233

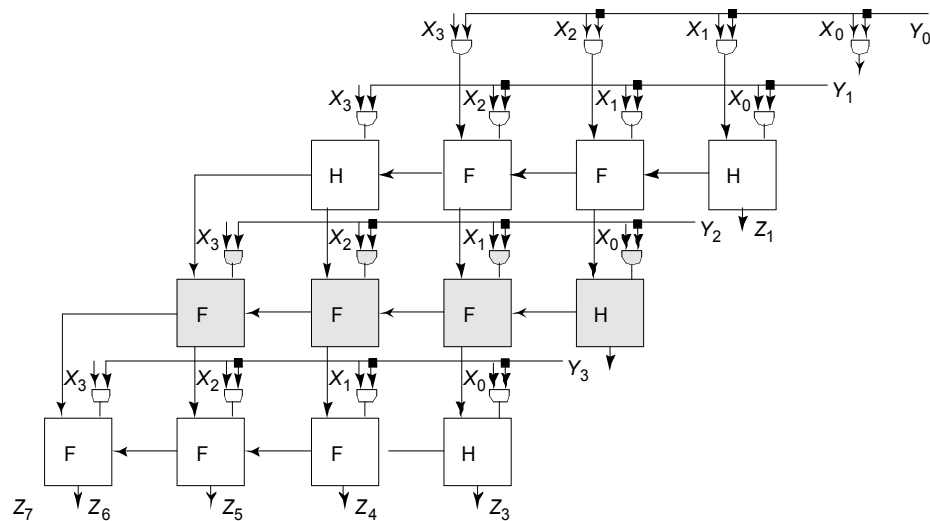
Multiplier

- X (M bits) and Y (N bits) unsigned
- Z (M+N bits)

$$\begin{aligned}
 Z &= X \times Y = \sum_{i=0}^{M+N-1} Z_i \times 2^i \\
 Z &= \left(\sum_{i=0}^{M-1} X_i \times 2^i \right) \times \left(\sum_{j=0}^{N-1} Y_j \times 2^j \right) \\
 Z &= \sum_{i=0}^{M-1} \left(\sum_{j=0}^{N-1} X_i Y_j \times 2^{i+j} \right)
 \end{aligned}$$

234

Array Multiplier



Critical Path: $T_{mult} =$

235

Booth Algorithm

- Exploit series of 1 or 0
- Suppress null partial products
- Simplifying property: $111 = 1000 - 1$

$$\begin{array}{r}
 01101 \\
 \times 01000 \\
 \hline
 00000 \\
 00000 \\
 00000 \\
 01101 \\
 \hline
 00000 \\
 0001101000
 \end{array}$$

$$\begin{array}{r}
 01101 \\
 \times 01000 \\
 \hline
 01101 \\
 00000 \\
 \hline
 0001101000
 \end{array}$$

$$\begin{array}{r}
 01101 \\
 \times 0111 \\
 \hline
 01101 \\
 01101 \\
 01101 \\
 00000 \\
 \hline
 001011011
 \end{array}$$

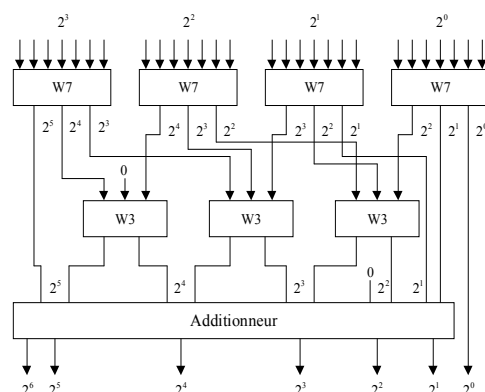
$$\begin{array}{r}
 01101 \\
 \times 0111 \\
 \hline
 01101000 \\
 - 01101 \\
 \hline
 001011011
 \end{array}$$

236

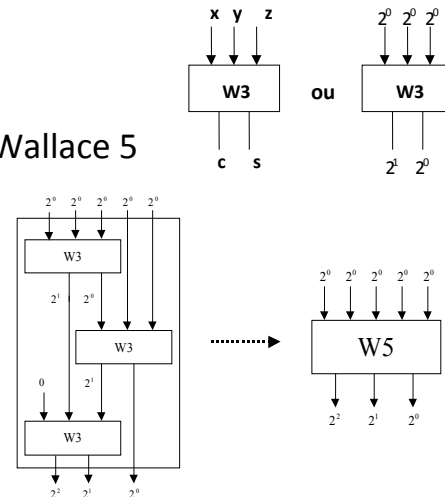
Wallace Tree

- Wallace 3

$$- S = X \oplus Y \oplus Z; C = XY + XZ + YZ$$



- Wallace 5



- Addition of seven 4-bit data

237

V Logic Synthesis from VHDL

1. Methods, Design Flow, and Tools
2. Register-Transfer Level (RTL) Models using VHDL
 1. VHDL for Logic Synthesis
 2. Combinational Logic
 3. Sequential Logic
3. RT and Logic Synthesis CAD Algorithms

Structuring, Flattening, Mapping

238

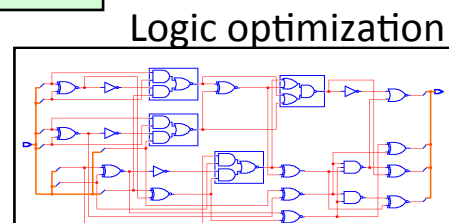
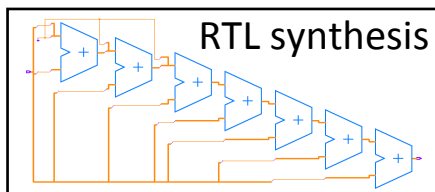
Logic Synthesis at a Glance

VHDL specification

```

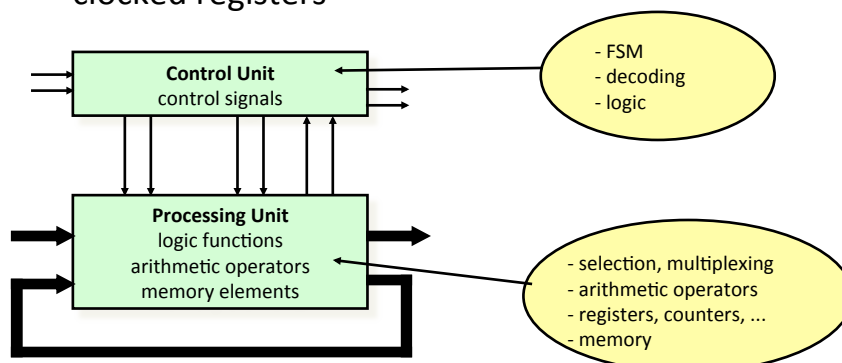
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.STD_LOGIC_ARITH.all;
entity nb2 is
    port(DIN: in std_logic_vector(7 downto 0);
         nb: out integer range 0 to 8);
end nb2;
architecture arch of nb2 is
begin
    process(DIN)
        variable nb_int: integer range 0 to 8;
    begin
        nb_int := 0;
        for i in 0 to 7 loop
            nb_int := nb_int + conv_integer(DIN(i));
        end loop;
        nb <= nb_int;
    end process;
end arch;
    
```

$$nb = \sum_{i=0}^7 DIN(i)$$

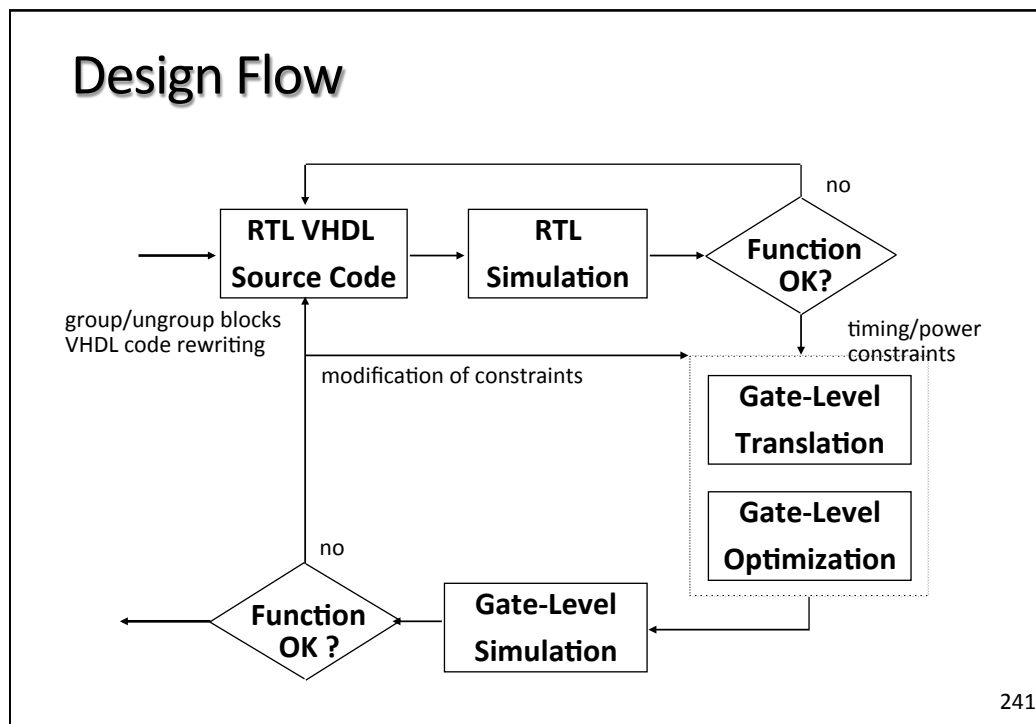


Logic Synthesis

- Register-Transfer Level (RTL) Specifications
 - Specification of the logic/arithmetic behavior between clocked registers



240



Advantages of Logic Synthesis (over gate-level design)

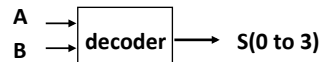
- Design flow automation
- Higher level of abstraction: more complex designs
- Hardware Description Language (HDL)
- Independent of technology (more or less...)
- Constrained design flow
 - timing, power, delay, area
- Optimized and analyzed results

242

This slide is to refresh your memory on VHDL

entity decoder is

```
port ( A, B : in bit;
      S  : out bit vector (0 to 3) );
end decoder ;
```



```
architecture behavioral
of decoder is
begin
  process (A,B)
  begin
    if A='0' then
      if B='0' then
        S<= "0001"
      else
        S<= "0010"
      end if;
    else
      if B='0' then
        S<= "0100"
      else
        S<= "1000"
      end if;
    end if;
  end process ;
end behavioral;
```

```
architecture dataflow
of decoder is
begin
  S(0) <= not(A) and not(B) ;
  S(1) <= not(A) and B ;
  S(2) <= A and not(B) ;
  S(3) <= A and B ;
end dataflow;
```

```
architecture structural
of decoder is
component DEC24
port ( I1, I2 : in bit;
      O1, O2, O3, O4 : out bit );
end component ;
begin
  CELL : DEC24
  port map (A,B,S(0),S(1),S(2),S(3));
end structural;
```

243

VHDL RTL coding style for synthesis

```
entity counter is
port (reset: in bit;
      S: out integer := 0);
end counter;

architecture behavioral of counter is
begin
  process
  variable count: integer := 0;
  begin
    wait until reset = '1' for 20 ns;
    if reset = '1' or count = 15
    then count := 0;
    else count := count + 1 after 10 ns;
    end if;
    S <= count;
  end process;
end behavioral;
```

Non Synthesizable

```
entity counter is
port (reset: in bit;
      clk: in bit;
      S: out integer range 0 to 15 );
end counter;

architecture RTL of counter is
signal count: integer range 0 to 15;
begin
  process(reset,clk)
  begin
    if reset='1' then count <=0;
    elsif clk'event and clk='1' then
      if count=15 then count <= 0;
      else count <= count + 1;
      end if;
    end if;
  end process;
  S <= count;
end RTL;
```

Synthesizable

244

VHDL RTL coding style for synthesis

Data types:	Integer (with range) Enumerate, Record, Subtype 1D array of finite dimension Bit, Bit_Vector ou STD_LOGIC, STD_LOGIC_VECTOR Physical, Real, Access, File: ignored
Entity:	<i>in, out, inout</i> , default values are ignored
Packages:	Collection of resources: types, constants, functions, components Standard packages (STD_LOGIC) or technology (design kit)
Declarations:	<i>Constant, Signal, Variable, Component</i> Register, Bus, Linkage, Alias: ignored
Operators:	Logic (and, nand, or, nor, xor, not) Comparison (=, /=, <, >, <=, >=) Arithmetic (+, -, *, sign, abs) (/, mod, rem, ** for a power of 2)
Attributes:	'length, 'event, 'left, 'right, 'high, 'low, 'range, 'reverse_range

245

VHDL RTL coding style for synthesis

Sequential Instructions (process)

Wait:	Supported at the first line of a synchronous PROCESS wait until clock = value; wait until clock'event and clock = value; wait until not clock'stable and clock = value;
Assignment:	Assignment of variables and signals: supported Functions and procedures: supported Transport after: ignored
If/Case:	Supported
Loops:	<i>for</i> loop with static size: supported <i>while</i> loop: not supported

Parallel Instructions (architecture)

Process:	Sensitivity list or wait for synchronization
Affectations:	Conditional assignment of signals (<i>when, select</i>): supported
Block:	Guarded blocks: not supported
Instantiation:	Port map, generic map, generate: supported

246

IEEE Standard Logic 1164 Package

```
PACKAGE std_logic_1164 IS
```

```
  TYPE std_ulogic IS (
    'U', -- Uninitialized
    'X', -- Forcing Unknown
    '0', -- Forcing 0
    '1', -- Forcing 1
    'Z', -- High Impedance
    'W', -- Weak Unknown
    'L', -- Weak 0
    'H', -- Weak 1
    '-' -- Don't care );
```

```
  TYPE std_ulogic_vector IS ARRAY ( NATURAL RANGE <> ) OF std_ulogic;
```

```
  FUNCTION resolved ( s : std_ulogic_vector ) RETURN std_ulogic;
```

```
  SUBTYPE std_logic IS resolved std_ulogic;
```

```
  TYPE std_logic_vector IS ARRAY ( NATURAL RANGE <> ) OF std_logic;
```

247

VHDL RTL coding style for synthesis

Conditional and parallel assignment of signals

```
S <= a+b;      S <= a and b      when a < "010"      with a select
                else a xor b when a < "101"          S <= a and b      when 2 downto 0,
                else a or b;                          a xor b      when 3 to 4,
                                                        a or b      when others;
```

Combinational Process

```
add: process (a,b,c)
begin
  if c='1' then res <= a + b;
  else          res <= a - b;
  end if;
end process;
```

- All **read signals** must be placed in the sensitivity list of the Process
- All **outputs** must be assigned for all possible values of the conditions

Synchronous Process

```
process (reset, clock) is begin
  if reset= '0' then
    -- actions during asynch. reset
  elsif clock'event and clock='1' then
    -- synchronous statements
    -- no possible else
  end if;
  -- no more possible actions
end process;
```

- Sensitivity list of the Process must contain the **clock** signal and eventually an **asynchronous signal** (reset)
- Sensitivity list can be replaced by a wait statement
- Outputs must be assigned both in synchronous and asynchronous manners

248

And then synthesis tools will do the job for you

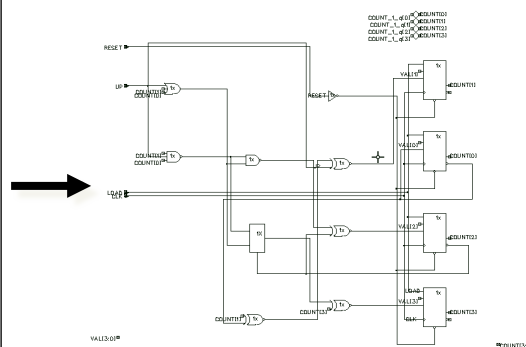
RTL Level Code

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity counter is
  port (reset, clk, load, up: in Std_Logic;
        val: in Std_Logic_Vector(3 downto 0);
        count : buffer Std_Logic_Vector(3 downto 0));
end counter;

architecture RTL of counter is
begin
  synchronous: process(reset, clk)
  begin
    if reset='1' then
      count <= "0000";
    elsif clk'event and clk='1' then
      if load = '1' then
        count <= val;
      elsif up = '1' then
        count <= count + "0001";
      else
        count <= count - "0001";
      end if;
    end if;
  end process;
end RTL;
  
```



Gate-Level Netlist

```

entity counter is
  port (...);
end counter;
architecture SYN_RTL of counter is
  component MUX21LL
    port (A, B, S: in std_logic; Z: out std_logic);
  end component;
  component FD2QLLP
    port (CD, CP, D: in std_logic; Q: out std_logic);
  end component;
  ...
end SYN_RTL;
  
```

```

...
begin
  ...
  U43 : MUX21LL port map( A => tqch, B => data, S => n90, Z => n89);
  U44 : MUX21LL port map( A => tich, B => tqch, S => n90, Z => n88);
  U45 : AN2LL port map( A => rstb, B => load_Fs, Z => n71);
  FF_regx0x : FD2QLLP port map( CD => rstb, CP => clk, D => n89, Q => tqch);
  FF_regx1x : FD2QLLP port map( CD => rstb, CP => clk, D => n88, Q => tich);
  ...
end SYN_RTL;
  
```

249

V Logic Synthesis from VHDL

1. Methods, Design Flow, and Tools
2. Register-Transfer Level (RTL) Models using VHDL
 1. VHDL for Logic Synthesis
 2. Combinational Logic
 - Logic gates
 - Multiplexer, tri-state
 - Arithmetic, comparison
 - Decoding
 3. Sequential Logic
3. RT and Logic Synthesis CAD Algorithms
 - Structuring, Flattening, Mapping

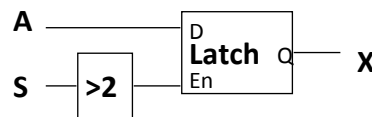
250

Be **very** Careful with Latches and Logic

Latch

```
process (A,S)
begin
  if S > 2 then
    X <= A;
  end if;
end process;
```

=>

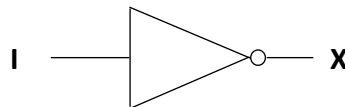


Logic gates

X <= not I

```
process (I)
begin
  if I = '0' then
    X <= '1';
  else
    X <= '0';
  end if;
end process;
```

=>



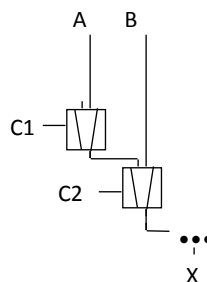
Complete specification of conditions to provide the full truth table

251

Multiplexers

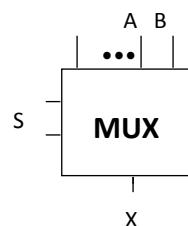
With priority

```
process (A,B,...C1,...)
begin
  if C1 then
    X <= A;
  elsif C2 then
    X <= B;
  ...
  else
    end if;
end process;
```



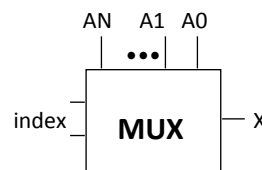
Without priority

```
process (A,B,...,S)
begin
  case (S) is
    when C1 =>
      X <= A;
    when C2 =>
      X <= B;
    ...
  end case;
end process;
```



Array Index

X <= A(index);



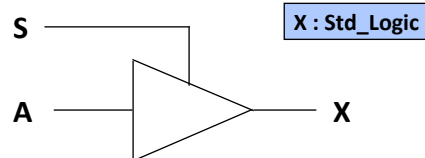
252

Tri-states and Loops

Tri-State

```
process (A,S)
begin
  if S = '1' then
    X <= A;
  else
    X <= 'Z';
  end if;
end process;
```

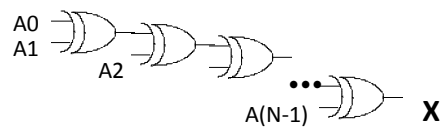
=>



Loop and Generate

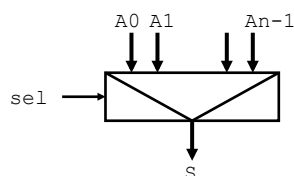
```
parité : process (A)
  variable result : bit;
begin
  result := '0';
  for i in 0 to N-1 loop
    result := result xor A(i);
  end loop;
  X <= result;
end process;
```

=>



253

Example: data selection

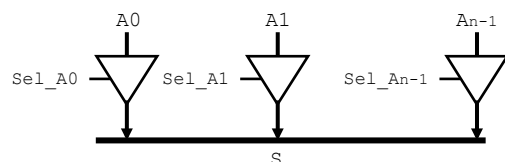
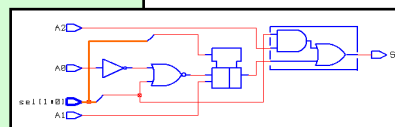


Multiplexer Logic

```
S <= A0 when sel=0 else
  A1 when sel=1 else
  A2 when sel=2 else
  'X';
```

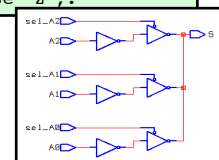
or

```
process(sel, A0,A1,A2)
begin
  case sel is
    when 0 =>
      S <= A0;
    when 1 =>
      S <= A1;
    when 2 =>
      S <= A2;
  end case;
end process;
```



Tri-state Logic

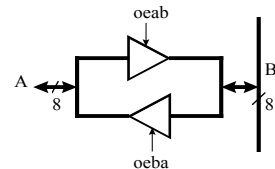
```
signal S, A0, A1,A2 : std_logic ;
signal Sel_A0,Sel_A1,Sel_A2 : std_logic;
...
S <= A0 when sel_A0='1' else 'Z';
S <= A1 when sel_A1='1' else 'Z';
S <= A2 when sel_A2='1' else 'Z';..
```



254

Example: Bidirectional Buffer

- Bidirectional buffer



```

library IEEE;
use IEEE.std_logic_1164.all;
entity transceive is
    port(

end entity
architecture RTL of transceive
begin

end RTL;

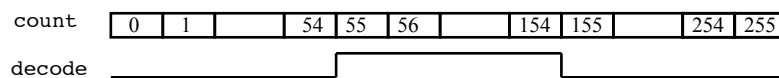
```

255

Example: Signal Decoding

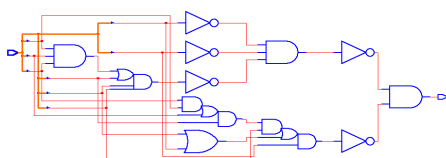
- Signal decoding

signal count: integer range 0 to 255;

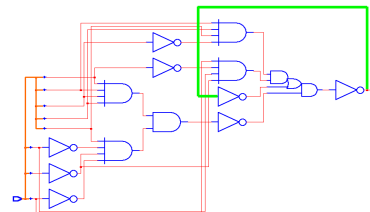


Two coding styles

Decoding of all possible values: Logic



Decoding of state transitions: Latch



256

Example: Signal Decoding

257

Arithmetic Operators

- Arithmetic operations

+ - * / (with restrictions)

$S \leq A \text{ op } B;$

- Signed

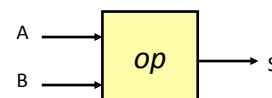
- integer range -128 to 127;
- signed(7 downto 0);

- Unsigned

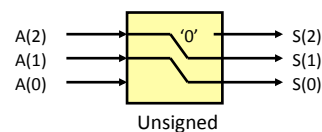
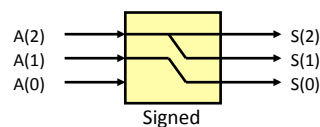
- integer range 0 to 255;
- unsigned(7 downto 0);

- Division: divider is a power of 2

$S \leq A/2;$



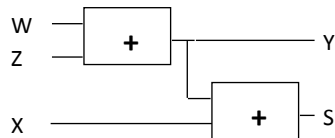
```
use IEEE.STD_LOGIC_1164.all;
use IEEE.STD_LOGIC_arith.all;
use IEEE.STD_LOGIC_SIGNED.all;
or
use IEEE.STD_LOGIC_UNSIGNED.all;
```



258

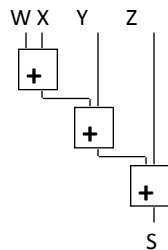
Parallel Statements and Assignements

```
architecture A of E is
begin
  S <= X + Y;
  Y <= Z + W;
end A;
```

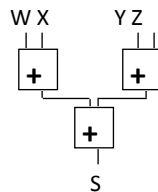


- Dataflow specifications
- Order of equations has no influence
- Combinational logic

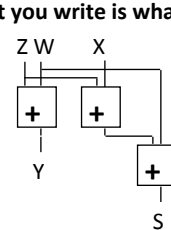
$S \leq W + X + Y + Z;$
Left to right evaluation



$S \leq (W + X) + (Y + Z);$
Effect of parentheses



$Y \leq Z + W;$
 $S \leq X + Z + W;$
What you write is what you have



259

V Logic Synthesis from VHDL

1. Methods, Design Flow, and Tools
2. Register-Transfer Level (RTL) Models using VHDL
 1. VHDL for Logic Synthesis
 2. Combinational Logic
 3. Sequential Logic
 - Flip-flops, registers
 - FSM
 - Memory
3. RT and Logic Synthesis CAD Algorithms

Structuring, Flattening, Mapping

260

Sequential Logic

- Edge-triggered flip-flop (FF) or registers
 - Sensitivity of the process is on the clock edge
 - value 'mem' could be a vector of bit/integer

```

register: process(clock)
  if (clock'event and clock = '1') then
    reg <= input_val;
  end if;
end process;

```

- Latch
 - Sensitivity of the process is on a signal level

```

latch: process(enable, input_val)
  if enable = '1' then
    mem <= input_val;
  end if;
end process;

```

261

Sequential Logic

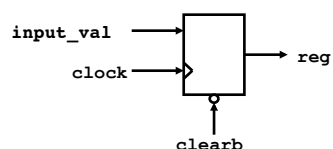
- Asynchronous or synchronous reset (or set)

asynchronous

```

process(clearb, clock)
  if clearb = '0' then
    reg <= 0;
  elsif (clock'event and clock = '1') then
    reg <= input_val;
  end if;
end process;

```

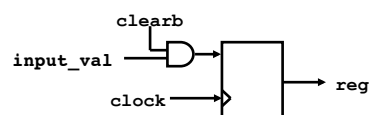


synchronous

```

process(clock)
  if (clock'event and clock = '1') then
    if clearb = '0' then
      reg <= 0;
    else
      reg <= input_val;
    end if;
  end if;
end process;

```



262

Variable or Signals?

Éléments de mémorisation

A signal assigned inside a [clk'event and clk='1'] statement will always be a direct output of a flip-flop

```

signal clk, A, B, S: std_logic;
.....
begin
...
process(clk)
begin
  if (clk'event and clk='1') then
    A <= B;
    S <= A;
  end if;
end process;

```

```

signal clk, B, S: std_logic;
.....
begin
...
process(horl)
  variable A: std_logic;
begin
  if (clk'event and clk='1') then
    S <= A;
    A := B;
  end if;
end process;

```

```

signal clk, B, S: std_logic;
.....
begin
...
process(clk)
  variable A: std_logic;
begin
  if (clk'event and clk='1') then
    A := B;
    S <= A;
  end if;
end process;

```

263

Example

Dessiner le schéma logique obtenu par synthèse de la description suivante :

```

process(clock)
  variable int1 : std_logic ;
begin
  if (clock'event and clock='1') then
    int1 := A nand B ;
    int2 <= C nor D ;
    S <= int1 nand int2 ;
  end if ;
end process ;

```

264

Some more rules

Only one clock and only one edge

```
process(clk)
  if (clk'event and clk = '1') then
    reg <= val1;
  end if;
  if (clk'event and clk = '0') then
    reg <= val2;
  end if;
end process;
```

Do not touch the clock!

```
process(clk)
  if (clk'event and clk = '1' and ena = '1') then
    reg <= val;
  end if;
end process;
```

Synchronous register with load

```
process(clk)
  if (clk'event and clk = '1') then
    if (ena = '1') then
      reg <= val;
    end if;
  end if;
end process;
```

265

Counting and Shifting

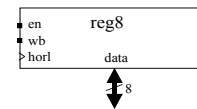
- Signal declared as
 - integer range 0 to N-1
 - signed/unsigned(n **downto 0**)
- Binary counter from 0 to N-1


```
if (compteur = N-1) then compteur <= 0;
else count <= count + 1;
end if;
```
- N-bit shift register
 - regdec(N-1 downto 1) <= regdec(N-2 downto 0);
 - regdec(0) <= '0';
 - or
 - regdec <= regdec/2;

266

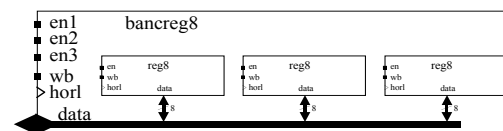
Example: 8-bit Register

- 8-bit register with bidirectional I/O and high-impedance output
- en = '1': read/write; en = '0': data = 'Z'
- en = '1' AND wb = '0': write
- en = '1' AND wb = '1': read
- data: 8-bit bidirectional I/O



267

Example: Register Bank



268

Memory

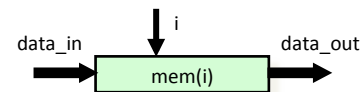
- A memory is a 1D-array of words

- Type declaration

```
type t_mem is array(0 to N-1) of std_logic_vector(nb_bits-1 downto 0);
type t_mem is array(0 to N-1) of integer range 0 to 2**(nb_bits-1);
```

- Signal declaration

```
signal mem: t_mem;
```



– $\text{mem}(i)$ is the i^{th} element of the memory

- Read the memory `data_out <= mem(i);`

- Write in the memory
 - edge or level triggered

```
if clk'event and clk='1' then
    mem(i) <= data_in;
end if;
```

269

Finite State Machines

- Enumeration of states

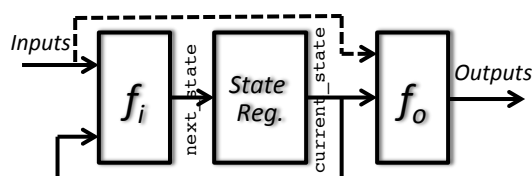
```
type states is (stat1, state2, state3, ...);
```

- State register declaration

```
signal current_state: states;
```

- State register

– synchronous process

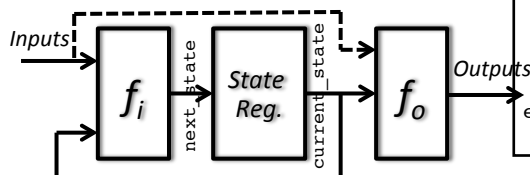


```
StateRegister: process(reset, clk)
begin
    if reset='0' then
        current_state <= init_state;
    elsif (clk'event and clk='1') then
        current_state <= next_state;
    end if;
end process;
```

270

Finite State Machines

- Decoding logic
 - combinational logic
 - remember to **respect RTL coding rules**
 - Moore/Mealy style
 - for Mealy outputs assignments are inside the condition on inputs

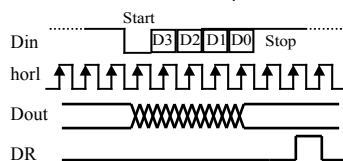


```
moore:process(inputs,current_state)
begin
  case current_state is
    when state1 =>
      outputs <= ...;
      if input1 = ...
      then next_state <= state2;
      else next_state <= state1;
      end if;
    when state2 =>
      outputs <= ...;
      if input2 = ...
      then next_state <= state5;
      else next_state <= state2;
      end if;
    ...
  end case;
end process;
```

271

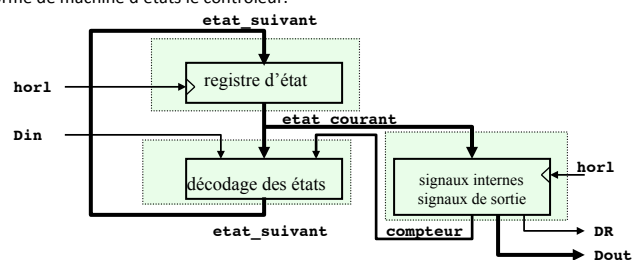
Exercice

Décrire la fonction “récepteur série asynchrone” de mots binaires de 4 bits

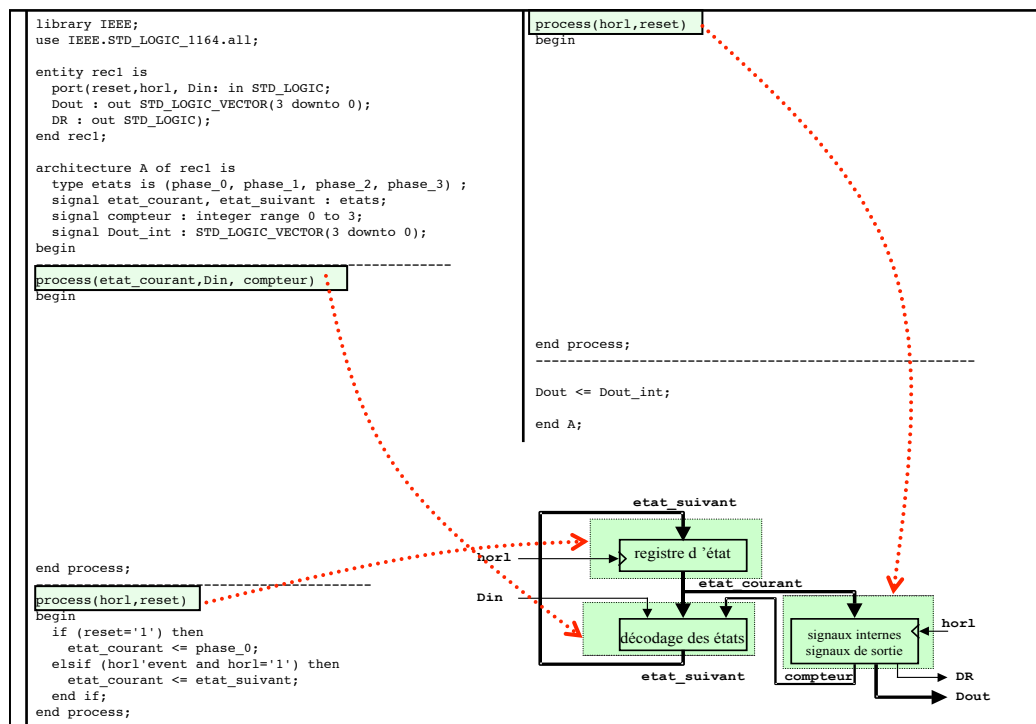


Le message binaire débute par un start bit (Din='0') suivi des 4 bits d'information et se termine par 2 stop bits (Din='1'). Le circuit reconstitue le mot de 4 bits et le présentera après réception sur le bus de sortie Dout en activant le signal DR (Data Ready).

Définir les différentes phases du traitement, attente du start bit, réception des 4 bits, et décrire sous forme de machine d'états le contrôleur.



272



V Logic Synthesis from VHDL

1. Methods, Design Flow, and Tools
2. Register-Transfer Level (RTL) Models using VHDL
 1. VHDL for Logic Synthesis
 2. Combinational Logic
 3. Sequential Logic
3. RT and Logic Synthesis CAD Algorithms

Structuring, Flattening, Mapping

Logic Synthesis Principle

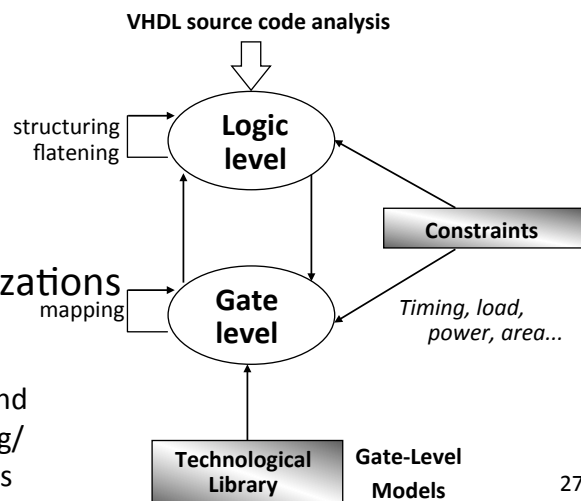
- Logic Synthesis = Translation + Optimizations

- Translation

- VHDL source code is analyzed and transformed into a generic logic representation

- Constrained optimizations

- Logic is mapped onto gates from the technological library and optimized under timing/power/area constraints



275

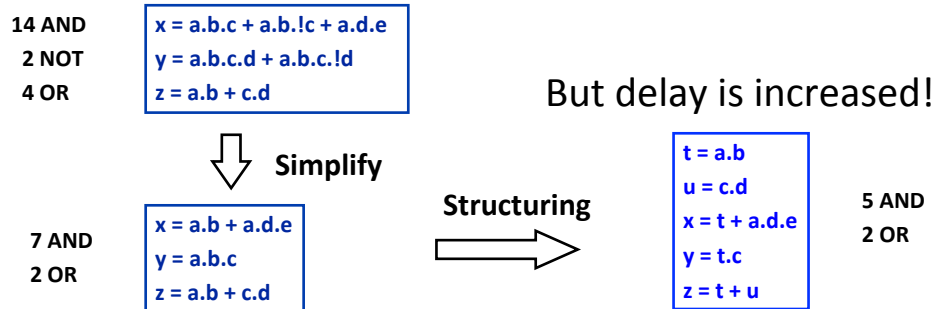
Optimizations

- Structuring: area oriented
 - Simplify and factorize Boolean equations
- Flattening: speed oriented
 - Distribute factors in Boolean equations
- Mapping
 - Selection of gates in the technological library
 - Specification of constraints (power/delay/load/area)
 - Generate several solutions

276

Optimizations

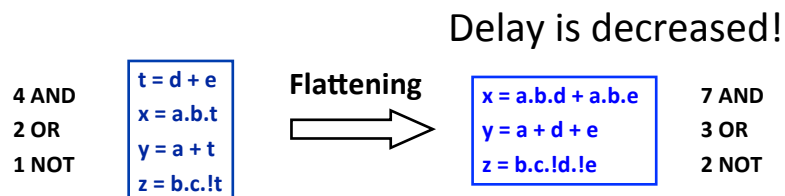
- Structuring: area oriented
 - Simplify and factorize Boolean equations
 - Share common factors



277

Optimizations

- Flattening: speed oriented
 - Distribute factors in Boolean equations
 - Reduces delay



278

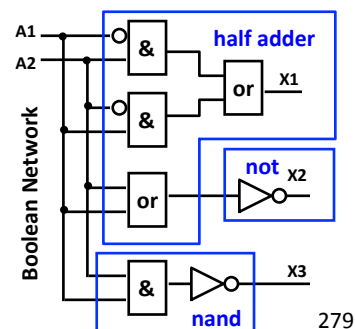
Optimizations

- Mapping

- Selection of gates in the technological library
- Specification of constraints (power/delay/load/area)
- Generate several solutions by graph covering

- Evaluate delay/cost
- Solution with lowest area respecting delay constraint is retained

Technological Library of Logic Gates (many gates in a 'design kit' library)			
gate	delay	cost	function
nand	2 ns	3	$\neg(A \cdot B)$
half adder	4 ns	14	$\neg A \cdot B + A \cdot \neg B$
••••			



279

Resource Sharing

```

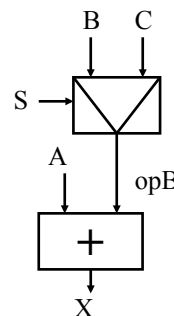
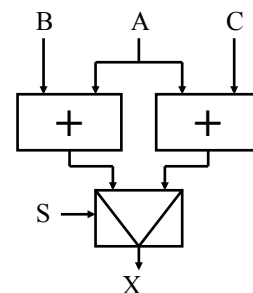
process (A,B,C,S)
begin
  if (S = ' 1 ' ) then
    X <= A + B ;
  else
    X <= A + C ;
  end if;
end process;

```

```

process (A,B,C,S)
  variable opB : integer;
begin
  if (S = ' 1 ' ) then
    opB := B ;
  else
    opB := C ;
  end if;
  X <= A + opB;
end process;

```



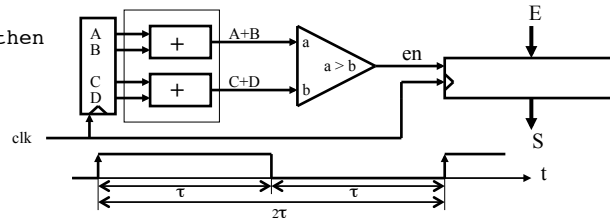
280

Pipeline

```

if (clk'event and clk = '1') then
  if (A + B) > (C + D) then
    S <= E;
  end if;
end if;

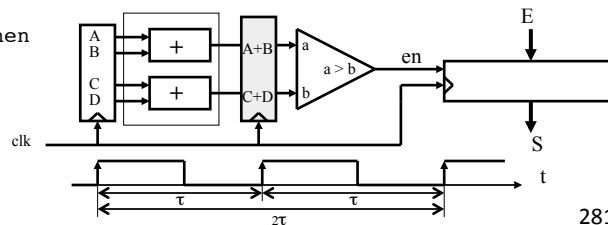
```



```

signal AB, CD: integer;
...
if (clk'event and clk = '1') then
  AB <= A + B;
  CD <= C + D;
  if AB > CD then
    S <= E;
  end if;
end if;

```



281